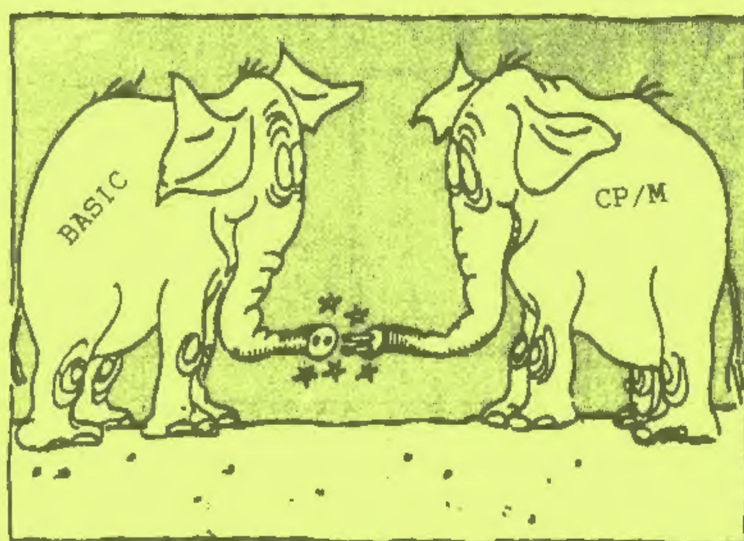
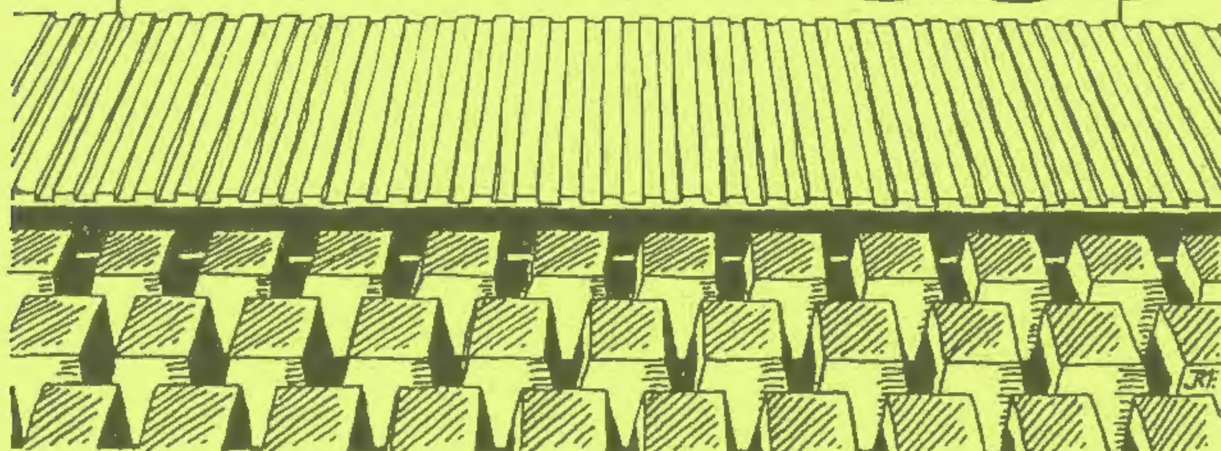
ATOM NIEUWS

JAARGANG
nummer





Bestuur:

Voorzitter:
Vacant

Secretaris:
J.Hartog
Keyenbergseweg 60
6871 WK Renkum
Tel.08373-13757

Penningmeester:
T.Rutten
Berkenlaan 24
3737 RN Groenekan
Tel.03461-3495

Clubwinkel:
P.Grevelt
Emmastraat 22
1782 PD Den Helder
Tel.02230-23453

Hardware cie.:
Vacant

Redactie Atom Nieuws:
H. de Ruiter
Polarisstraat 25
8303 AC Emmeloord

Contributie 1988: fl. 50.00
Leden buitenland: fl. 75.00

Rekeningnummers Atom Computer Club:
Giro 5244293

Redactie Atom Nieuws:
Joop Ballijns
Harry de Ruiter

Ledenadministratie:
S.van Leeuwen
Kompasstraat 32
1973 PX IJmuiden
Tel.02550-22435

Uiterste datum inlevering copy:
nr. 88-1   01-03-1988

De clubwinkel:

80-koloms videokaart, incl.alle onderdelen behalve de RAM-IC's, bouwpakket	fl.130.00
80-koloms videokaart,gebouwd en getest,excl.RAM-IC's	fl.180.00
Geheugenkaart; 16 kByte,excl.onderdelen	fl. 35.00
Schakelkaart;	niet meer leverbaar
Minischakelkaart	fl. 16.00
Z-80 kaart;CP/M,excl. onderdelen	fl.100.00
Herdruk ACORN NIEUWS 1982; 97 pag.wetenswaardigheden	fl. 6.00
ATOM NIEUWS jaargang 1983, ruim 450 pag.	fl. 30.00
ATOM NIEUWS jaargang 1984, ruim 650 pag.	fl. 35.00
ATOM NIEUWS jaargang 1985, ruim 650 pag.	fl. 35.00
ATOM NIEUWS Jaargang 1986, ruim 500 pag.	fl. 35.00
ATOM-WARE 1; Basic interpreter van de ATOM. 98 pag.	fl. 6.00
ATOM-WARE 2; ATOM Disc Operating System,68 pag.	fl. 5.00
ATOM-WARE 3; ATOM Monitor Operating System,80 pag.	fl. 5.00

Levering;
Bij uw reg.penningmeester, eventueel rechtstreeks bij de fed. penningmeester.Bij rechtstreekse bestelling dient u het bedrag van het gewenste artikel te storten op de giro van de federatie onder vermelding van de naam van het artikel en uw lidnummer.Uw betaling dient vermeerderd te zijn met fl. 4.00 voor portokosten.

pag 2	uit de federatie	
pag 3	inhoudsopgave/mededeling	
pag 4	CP/M autostart	Nico van Rooyen
pag 5-9	65816 nader bekeken	
pag 10-11	65816 in ATOM	
pag 12-13	NMOS met MINIAS	A.M.Vink
pag 14-16	MIX	Guido de Roos
pag 17-19	Archimedes	Acorn
pag 20-24	Disc-menu	Guido de Roos
pag 25-30	Storingen	Joop Ballijns
pag 31-35	Teletekst	Bas Kasteel
pag 36-37	Niemandalletje	Theo Waayer
pag 38-43	Electronica	Jan Bronzwaer
pag 44-48	Automaire wetenswaardigheden	G.v/d Brandhof
pag 49	Bus problemen	Sjaak Geene
pag 50	Sneeuw	Jan Wijnen
pag 51	Klok	
pag 52-53	Exec.stat	Hans Hegt
pag 54	MDCR redmiddel	Jan Wijnen
pag 55	BB/BO	
pag 56-58	D/A converter	Jan Wijnen
pag 59	Eproms	Sjaak Geene
pag.60	Regionale adressen	

Mededeling van de penningmeester.

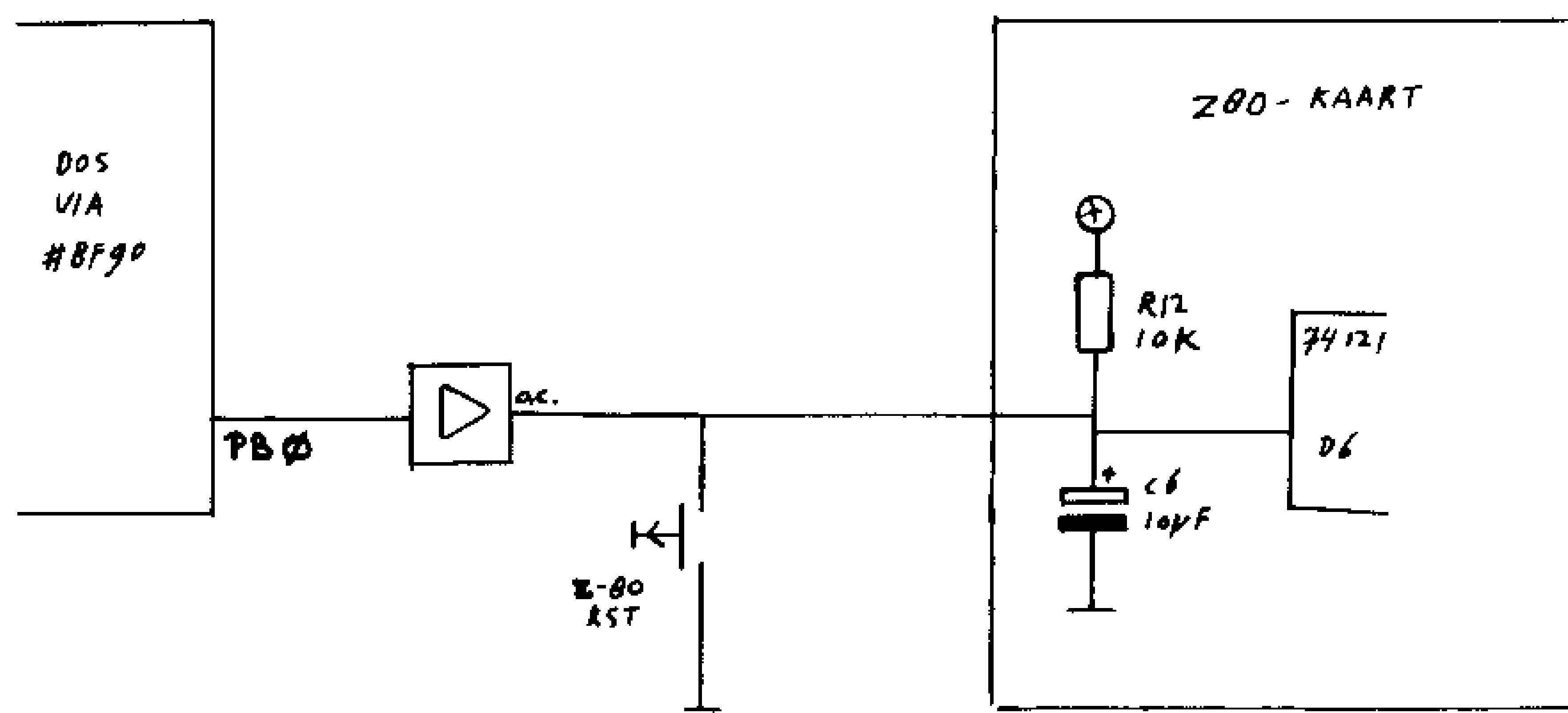
Aangezien u alweer het laatste nummer van ATOM NIEUWS van dit jaar in uw handen hebt treft u zoals gewoonlijk een accept-girokaart aan. Ook diegene die te kennen heeft gegeven het lidmaatschap te willen beeindigen. Voor ieder zal duidelijk zijn dat de automatisering hiervoor verantwoordelijk is. In verband met het verzenden van ATOM NIEUWS wordt u verzocht de contributie voor 13 februari 1988 over te maken. Na deze datum betalen betekent extra verzendkosten. Deze worden dan ook in rekening gebracht.

ATOM NIEUWS is een uigave van de federatie ATOM computerclubs Ned/Belgie en verschijnt 4 keer per jaar.

CP/M AUTO-START BOX

Gebruikt u wel eens CP/M programma's?
In dat geval heeft u natuurlijk ook het terminal-programma in uw ATOM nodig. Aangezien ik er genoeg van kreeg om steeds weer het terminal-programma te laden en daarna te runnen, heb ik het terminal-programma in een box op #A000 gezet.
Het terminal-programma gebruikt echter RAM achter zijn object-code, zodat het in ROM plaatsen voorlopig nog niet aan de orde is.
Daarom zit er een copieer-routine in de box ingebouwd die een copie van het terminal-programma overbrengt naar het geheugengebied vanaf #8200. Verder zit er in de box opgenomen een statement "herkenning" volgens het programma zoals beschreven in ATOM NIEUWS 3.3 blz.64. Ook wordt in het programma bit BPO van de VIA 6522 even laag gemaakt. Hieraan is door middel van een open collector aansluiting de reset van de Z-80 kaart aan te sluiten. (zie onderstaand schema).
Het is nu mogelijk om vanuit Basic via het commando CPM direct de Z-80 kaart te resetten en CP/M op te starten.

Nico van Rooyen



*** 65816 NADER BEKEKEN ***

In het overzicht van de instructie-set zijn al aardig wat zaken aan bod gekomen, maar nog lang niet alles. Vandaar dit vervolg met onder meer aandacht voor hardware aspecten, interrupt mogelijkheden en adresserings eigenschappen.

HARDWARE KENMERKEN

65802/65816 - CMOS technologie (laag vermogen, hoge storingsdrempel)
- interne 16-bits structuur (16 bits mogelijkheid in native mode)
- bidirectionele RDY-lijn (was input bij 6502)
- klok-frequenties 2,4,6,8 MHz

65802 - pin-compatible met 6502 (adresbereik 64Kbyte)

65816 - niet pin-compatible met 6502 (zie aansluitingen)
- adresbereik 16 Mbyte (middels 8 extra adreslijnen)
- 8 nieuwe control-lijnen (ten koste van 4 oude control-lijnen)

Aansluitingen:

(VSS)	$\overline{\text{VP}}$	1	40	$\overline{\text{RES}}$	
	RDY	2	39	VDA	(PHI2 out)
(PHI1 out)	$\overline{\text{ABORT}}$	3	38	M/X	($\overline{\text{SO}}$)
	$\overline{\text{IRQ}}$	4	37	PHI2 in	
(NC)	$\overline{\text{ML}}$	5	36	BE	(NC)
	$\overline{\text{NMI}}$	6	35	E	(NC)
(SYNC)	VPA	7	34	R/$\overline{\text{W}}$	
	VDD	8	33	D0/A16	(D0)
	A0	9	32	D1/A17	(D1)
	A1	10	31	D2/A18	(D2)
	A2	11	30	D3/A19	(D3)
	A3	12	29	D4/A20	(D4)
	A4	13	28	D5/A21	(D5)
	A5	14	27	D6/A22	(D6)
	A6	15	26	D7/A23	(D7)
	A7	16	25	A15	
	A8	17	24	A14	
	A9	18	23	A13	
	A10	19	22	A12	
	A11	20	21	VSS	

A16..A23 = Bank address (outputs)
De 65816 is in staat om direct 16 Mbyte adresruimte te adresseren. Dit zijn 256 blokken van 64 Kbyte. Zo'n blok van 64 Kbyte noemen we een bank. De adreslijnen A16..A23 bepalen welke bank geselecteerd wordt. Deze 8 extra adreslijnen worden gemultiplexed met de databus (t.b.v. 40 pens behuizing). Dat wil zeggen dat de pennen 26..33 om de beurt dienst doen als databus en bank-adresbus. Tijdens de 1e helft van de klokperiode (phi2) is het bank-adres beschikbaar en tijdens de 2e helft wordt de bus als databus gebruikt. Om daadwerkelijk van het bank-adres gebruik te kunnen maken, moet dit in een externe latch geklokt worden.

ABORT = abort (input)
Een negatieve flank op deze ingang resulteert in een speciale interrupt. Hierbij worden alle interne registers geblokkeerd zodat deze door de in uitvoering zijnde instructie niet kunnen worden gewijzigd. Nadat de lopende instructie als een soort NOP is afgemaakt, wordt een interrupt sequence afgehandeld. Het adres van de opcode van de afgebroken instructie wordt daarbij als return-adres op de stack geplaatst. Daarna wordt de PC geladen met de ABORT vector en vervolgt de processor zijn weg met de ABORT service routine.

BE = Bus Enable (input)
Met een laag nivo op deze ingang kan men de adresbus, databus en R/W-lijn van de processor in hoogohmige toestand brengen.

E = Emulation status (output)
Deze uitgang reflecteert de toestand van de E-flag in het status register (P).

ML = Memory Lock (output)
Bij Read-Modify-Write instructies is dit signaal laag gedurende de laatste 3 of 5 klokperioden (afhankelijk van de toestand van de M-flag).

M/X = Memory/indeX status (output)
Deze gemultiplexte uitgang reflecteert tijdens de negatieve flank van phi2 de toestand van de M-flag en tijdens de positieve flank de toestand van de X-flag.

VDA = Valid Data Address (output)
VPA = Valid Program Address (output)
Deze 2 uitgangen geven aan welk type geheugen aangesproken wordt. Codering:

VDA	VPA	
0	0	interne operatie (adres- en databus beschikbaar)
0	1	geldig programma adres (ophalen 2e, 3e, 4e instructie-bytes)
1	0	geldig data adres (lezen/schrijven van data)
1	1	opcode fetch (ophalen 1e instructie-byte)

VP = Vector Pull (output)
Deze uitgang geeft aan dat een vector locatie geadresseerd wordt tijdens een interrupt (is laag gedurende de laatste 2 interrupt klokperioden).

Vrijwel alle bovengenoemde control-signalen zijn bedoeld voor gebruik in grotere systemen met zaken als co-processor, DMA (Direct Memory Acces), virtueel geheugen, cache geheugen e.d.

INTERRUPTS

Hardware interrupts: ABORT, IRQ, NMI, RES (inputs)
Software interrupts: BRK, COP (instructions)

Tijdens een interrupt afhandeling vinden de volgende acties plaats:

```
PB -> stack  (vervalt in emulatie mode)
PC -> stack  (PCH eerst)
P  -> stack  (B-flag=0 op stack bij hardware interrupt in emulatie mode)
1 -> I-flag, 0 -> D-flag
00 -> PB
vector -> PC
```

Vector locaties:

int	E=1	E=0
IRQ	00FFFE,F	00FFEE,F
BRK	00FFFE,F	00FFE6,7
RESET	00FFFC,D	00FFFC,D
NMI	00FFFA,B	00FFEA,B
ABORT	00FFF8,9	00FFE8,9
COP	00FFF4,5	00FFE4,5
reserve	00FFF6,7	00FFEC,D

Initialisatie bij RESET:

registers	flags
D =0000	M=1
DB=00	X=1
PB=00	D=0
SH=01	I=1
XH=00	E=1
YH=00	

ADRESSEERBARE PROGRAMMA-GEHEUGEN (65816)

Het adresseerbare programma geheugen is niet "continu" over 16 Mbyte. D.w.z. dat er een "wrap-around" modulo 64K optreedt. Dus als de PC van #FFFF "verhoogd" wordt naar #0000, dan gaat de processor verder aan het begin van dezelfde geheugenbank. M.a.w. het PB register is niet aan het PC register gekoppeld. Het PB register kan alleen gewijzigd worden door interrupts en door de volgende instructies: JML, JSL, RTL, RTI, BRK en COP.

ADRESSEERBARE DATA-GEHEUGEN (65816)

Het adresseerbare data geheugen is wel continu over 16 Mbyte. D.w.z. dat bij ge-indexeerd adresseren het effectief adres door kan lopen over een bankgrens. Direct-, stack-, en vector-adressering hebben altijd betrekking op bank 0.

DIRECT ADRESSERING

```
E=1   DL=00  bereik 00DH00..00DHFF wrap-around mod 256 (1)
      DL<>00 bereik 00DHDL..00DHDL+1FE continu binnen bank 0.

      (1) enkele instructies kunnen in de volgende pagina doorlopen.
      Dit zijn PEI en instructies met de adresseringsvormen [d] en [d],y.

E=0   bereik 000000..00FFFF continu binnen bank 0 (wrap-around mod 64 K)
```

STACK ADRESSERING

E=1 bereik 000100..0001FF wrap-around mod 256
Er zijn enkele instructies die deze grenzen kunnen overschrijden.
Dit zijn: JSL, JSR (a,x), PEA, PEI, PER, PHD, PLD, RTL en instructies met de adresseringsvormen r,s en (r,s),y.

E=0 bereik 000000..00FFFF continu binnen bank 0 (wrap-around mod 64 K).

BLOCK-MOVE ADRESSERING

X=1 bereik sb,0000..sb,00FF en db,0000..db,00FF wrap-around mod 256.

X=0 bereik sb,0000..sb,FFFF en db,0000..db,FFFF wrap-around mod 64 K binnen de gekozen banks.

sb = bron bank
db = bestemmings bank
X = bron adres (8 of 16 bits afhankelijk van X-flag)
Y = bestemmings adres (8 of 16 bits afhankelijk van X-flag)
C = bloklengte -1 (altijd 16 bits onafhankelijk van M-flag)

16-BIT DATA ADRESSERING

De nieuwe processoren hebben nog steeds een 8-bits databus en het geheugen is nog steeds 8 bits breed. 16 bits data heeft dan ook betrekking op 2 opeenvolgende geheugenplaatsen met lage byte eerst. De 2 bytes worden na elkaar getransporteerd hetgeen 2 klokperioden in beslag neemt.

PROCESSOR STATUS REGISTER

```
E=1         M X       E    M=1 X=1 (bit4 en bit5 kunnen in emulatie mode
          N V 1 B D I Z C  niet gewijzigd worden met REP/SEP)


E=0         1 B       E    (BRK en IRQ hebben in native mode ieder een eigen
          N V M X D I Z C  vector dus B-flag is nu overbodig)
```

In beide modes geldt dat PHP, PLP, REP en SEP alleen betrekking hebben op het "onderste" deel van het status register. De E-flag is alleen bereikbaar via de XCE instructie. Als X-flag=1, geldt altijd: XH=00 YH=00.

EMULATIE MODE VERSUS NMOS 6502

functie	emulatie mode	NMOS 6502
Decimal mode: - na reset	D-flag=0	D-flag is ongedefinieerd
- na interrupt	D-flag=0	D-flag is ongewijzigd
- N,V,Z flags	geldig	ongeldig
JMP (a): als a=XXFF	correct: (a,a+1) -> PC	foutief: (a,a-FF) -> PC
Direct adressering: d	adres=D+d	adres=0+d (zp)
Absoluut adressering: a	adres=DB,a	adres=00,a (abs)

EMULATIE MODE VERSUS NATIVE MODE

functie	emulatie mode	native mode
Stackpointer	8 bits (SH=01)	16 bits
Direct adressering ge-indexeerd	wrap-around mod 256	continu
branch over pagina-grens	4 cycles	3 cycles
Processor status (P): - bit4	B-flag	X-flag
- bit5	altijd 1	M-flag
Vector locaties: - ABORT	00FFF8,9	00FFE8,9
- BRK	00FFFE,F	00FFE6,7
- COP	00FFF4,5	00FFE4,5
- IRQ	00FFFE,F	00FFEE,F
- NMI	00FFFA,B	00FFEA,B
Program bank bij interrupt/RTI	PB niet naar/van stack	PB wel naar/van stack

RECTIFICATIE

In het overzicht van de instructie-set van de vorige keer zijn enkele foutjes geslopen. Bij de TCS instructie worden de N en Z flags niet beinvloed. In emulatie mode is de benaming van de TCS instructie enigzins misleidend want alleen AL wordt naar SL getransporteerd. Als E=1 geldt immers altijd dat SH=01. Bij de adresseringsvorm al,x is het aantal klokperioden in 16 bits mode 6. Bij de adrsseringsvormen (d),y [d],y a,x a,y al,x en (r,s),y kan het gegenereerde bank-adres een bank hoger zijn dan vermeld omdat het data-geheugen continu is. In emulatie mode is het adresbereik niet beperkt tot bank 0. D.m.v. de lange adresseringsvormen is dus ook vanuit emulatie mode het hele 16 Mbyte geheugen bereikbaar.

Bronnen: - data sheet G65SC802/G65SC816 GTE
- eigen waarnemingen met G65SC816

*** 65816 IN ATOM ***

In het eerste nummer van 1986 sprak Jaap v/d Veen nog over toekomstdromen. Hoe deze dromen werkelijkheid kunnen worden, leest u in het nu volgende artikel.

65802 IN ATOM

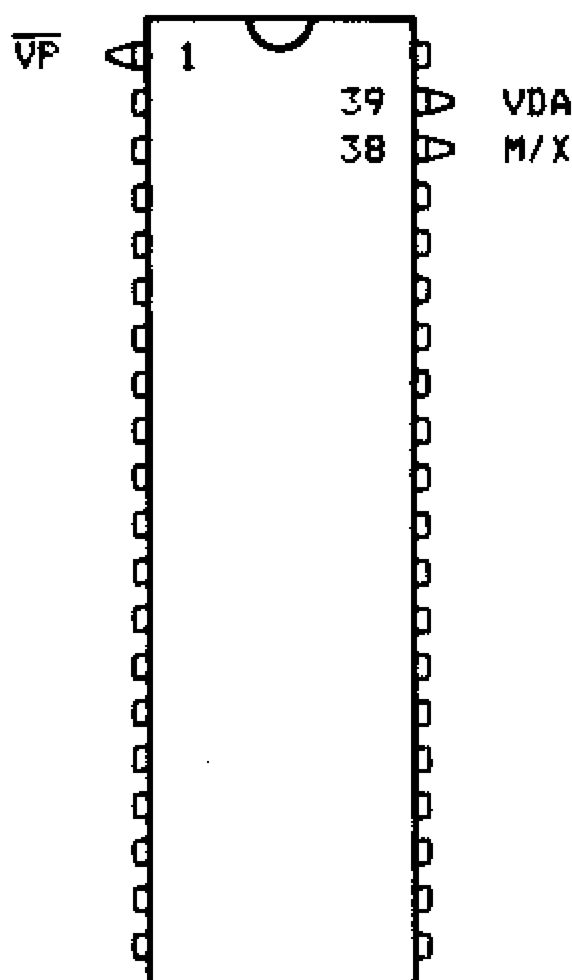
Hierover kan ik kort zijn: 6502 (ic22) eruit, 65802 erin, klaar!
Dit is dus erg gemakkelijk, maar misschien niet zo verstandig. We blijven met een 64K geheugen zitten zonder vooruitzichten op een wezenlijk grotere capaciteit.

65816 IN ATOM

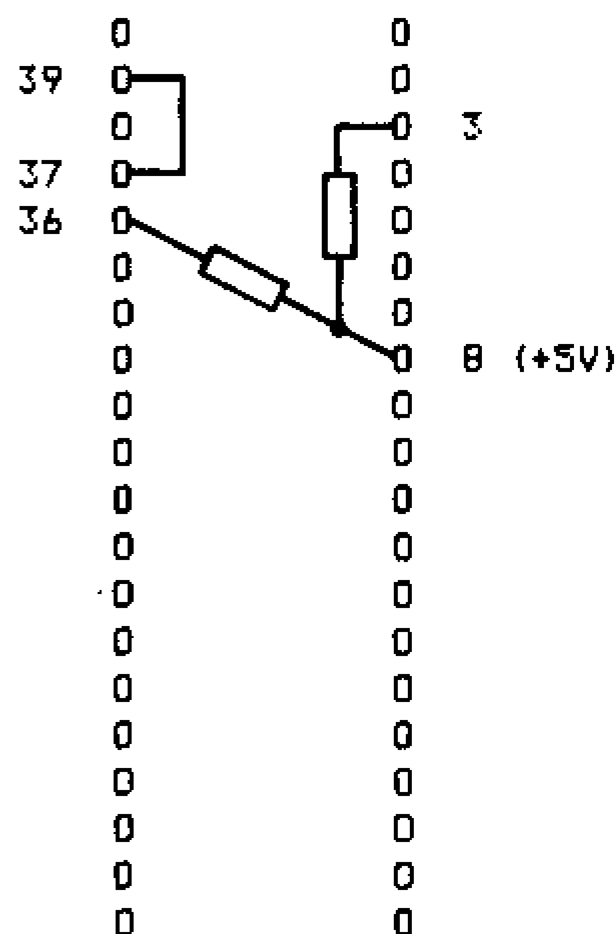
Als we nu zonder al te veel wijzigingen de ATOM geschikt kunnen maken voor de 65816 processor, dan hebben we wel de mogelijkheid om later het geheugen uit te breiden. Welnu, het bleek te kunnen met 2 extra weerstandjes en een stukje draad.

pen	65816 CPU		ic-voet 22		
1	$\overline{\text{VP}}$ out	--//--	GND		$\overline{\text{VP}}$ pen uitbuigen
3	$\overline{\text{ABORT}}$ in	------	PHI1 out	3K9 +5V	$\overline{\text{ABORT}}$ met 3K9 aan +5V leggen
5	$\overline{\text{ML}}$ out	------	NC		
7	VPA out	------	SYNC out		VPA is als SYNC te gebruiken
35	E out	------	NC		
36	BE in	------	NC	3K9 +5V	BE met 3K9 aan +5V leggen
38	M/X out	--//--	$\overline{\text{SO}}$ in		M/X pen uitbuigen
37	PHI in	------	PHI in	┐	PHI in met PHI2 out verbinden
39	VDA out	--//--	PHI2 out	┘	VDA pen uitbuigen

Haal eerst de 6502 eruit. Soldeer vervolgens de 2 weerstanden en de doorverbinding op de print aan de koperzijde (side one). Buig nu voorzichtig de pootjes 1, 38 en 39 van de 65816 iets naar buiten. Plaats nu de 65816 zodanig in ic-voet 22 dat de uitgebogen pennen net naast het voetje steken.



bovenaanzicht 65816



aanzicht koperzijde ic22

VECTOR LOCATIES

De nieuwe vector locaties hebben helaas conflicten met het ATOM operating system tot gevolg. In emulatie mode betreft dit alleen de COP vector en bij de 65816 ook de ABORT vector. In native mode is het nog erger want dan vallen alle vectoren (behalve RESET) samen met belangrijke JMP adressen uit het operating system. De vectoren wijzen daardoor noodgedwongen naar de meest vreemde locaties:

```
native mode:  COP --> #020A     emulation mode:  COP --> #086C
              ABR --> #C9FF                      ABR --> #0206
              BRK --> #E320
              IRQ --> #200A
              NMI --> #D00D
```

Deze faciliteiten zijn dus eigenlijk onbruikbaar en dat is een behoorlijke beperking. Bij de 65816 is daar wel wat aan te doen, maar dit vergt wel extra hardware. Voorlopig stellen we het dus maar zonder interrupts in native mode en zonder COP in emulatie mode.

EXTRA GEHEUGEN (65816)

Voor extra RAM en/of ROM in de hogere geheugen banken is eveneens extra hardware nodig. Of dit eenvoudig te realiseren is, hangt af van de mate waarin de betreffende ATOM reeds "opgevoerd" is en hoe dit gedaan is. Het lijkt mij het beste om daar in een apart artikel nog eens op terug tekomen. Voorlopig houden we het dus maar bij de bestaande 64K in bank 0.

KOSTEN

De firma Meek-it electronica in Den Haag kan de 65816 leveren voor 95 gulden. Dit betreft dan een 4 MHz versie van het fabrikaat GTE. De 65802 kost eveneens 95 gulden, maar is slecht leverbaar. Echt goedkoop zijn ze dus niet, maar dat schijnt een familie-kwaal van de 65-serie te zijn.

GEBRUIK

Over het programmeren van de 65816 valt erg veel te vertellen maar ik wil me hier beperken tot enkele praktische aanwijzingen. Zorg dat de I-flag 1 is voordat je overschakeld naar native mode. Omschakelen van emulatie naar native mode gaat eenvoudig met CLC;XCE. De flags M en X veranderen hierdoor niet, dus het uitgangspunt is altijd 8-bits mode. Met REP M,X of REP M of REP X kan men vervolgens naar de gewenste 16-bits mode omschakelen. Terug naar E-mode gaat met SEC;XCE waarbij de M en X flags automatisch 1 worden.

Wees erop bedacht dat de instructies ORA AND EOR ADC STA CMP SBC geen d,y adresserings mode kennen. De volgende 2 instructies adresseren alleen dezelfde locatie als het D-reg en DB-reg beiden 0 zijn:

```
LDA #80,Y  -->  24 bits eff adres  =  DB,#0080+Y
LDX #80,Y  -->  24 bits eff adres  =  00,D+#80+Y
```

Verder is het niet aan te raden om vanuit N-mode 6502 subroutines aan te roepen, zelfs niet als D=0000, DB=00, M=1 en X=1. Er treedt in N-mode namelijk geen wrap-around effect op bij ge-indexeerd adresseren van de direct page. Sommige routines maken daar met opzet gebruik van, zoals bijv. #F7F1.

Inleiding

Na in AN5.6 het artikel van J. Jobse gelezen te hebben over de CMOS/NMOS monitor heb ik de NMOS versie maar eens omgebouwd naar een MINIAS versie en gekeken of deze monitor werkelijk handiger is dan de bekende MONITOR.RH .
En hier volgt dan de conclusie.

De Monitor

Als eerste moet ik zeggen dat ik zelf gebruik maak van een ELEKTUUR 80 kolomskaart met een zelfgeschreven driver, dus na het slotwoord gelezen te hebben vroeg ik me af hoelang ik de monitor op het 32 kolomsscherm zou gebruiken want de breakpoint mogelijkheden en het display-point bevielen me eigenlijk wel.
Wat ik eigenlijk miste was de mogelijkheid om labels op te nemen in de step/trace en disas mogelijkheid, dus hier heb ik ook wat aan gedaan waardoor een labeltabel die in het geheugen staat in de disas of step/trace lijst wordt opgenomen.
Tevens is de uitvoer naar het beldscherm aangepast naar een 80 kolomsscherm.
De uitvoer van een trace zal er dan alsvolgt uitzien:

```
LABEL      ADD  HEXCODE   INSTRUCTION     PC   A  X  Y  S  P NV BDIZC  R

zero       9BD5 A9 00     LDA @#00        9BD5 45 2A 00 FF 22 00100010 F8
           9BD7 8D 3D 01  STA areg        9BD7 00 2A 00 FF 22 00100010 F8
           9BDA 8D 3C 01  STA xreg        9BDA 00 2A 00 FF 22 00100010 F8
           9BDD 8D 3B 01  STA yreg        9BDD 00 2A 00 FF 22 00100010 F8
           9BE0 A9 34     LDA @#34        9BE0 00 2A 00 FF 22 00100010 F8
           9BE2 8D 39 01  STA preg        9BE2 34 2A 00 FF 20 00100000 F8
reset'sreg 9BE5 A9 FF     LDA @#FF        9BE5 34 2A 00 FF 20 00100000 F8
```

Door zo'n uitvoer is aan de hand van een source listing een programma veel gemakkelijker te volgen.
Mochten er twee tabellen in het geheugen staan dan wordt alleen de hoogste gevonden waarbij naar een tabel gezocht wordt vanaf #9F00 naar beneden tot #2000.
Als er geen tabel gevonden word, dan zal de monitor normaal de adressen weergeven.
Ook is de weergave van het HEX commando verandert. de weergave is in principe gelijk gebleven alleen worden nu 16 adressen op een regel weergegeven waarna de ASCII tekens komen.

Gevolgen en Consequenties

Het gevolg van de uitbreiding is dat de object code nu exact 2K is waardoor hij nog net in het stuk geheugen van 7800-7FFF of van 9800-9FFF past.
Een ander gevolg van de uitbreiding is dat ik twee extra zeropage adressen heb moeten gebruiken (dit zijn de adressen #A7 en #A8 geworden) en ik heb 5 Absolute adressen gebruikt die bij mij in het rommelram gedeelte van #400 tot #7FF liggen, maar hiervoor is elk ander adres te gebruiken.
Om het TRACE commando zo duidelijk te programmeren heb ik gebruik gemaakt van twee routines die in mijn VDU-80 driver zitten, namelijk:

1. Clear to end of line DC1 = ctrl-Q (#11) regel 6590
2 Clear to end of screen DLE = ctrl-P (#10) regel 5270

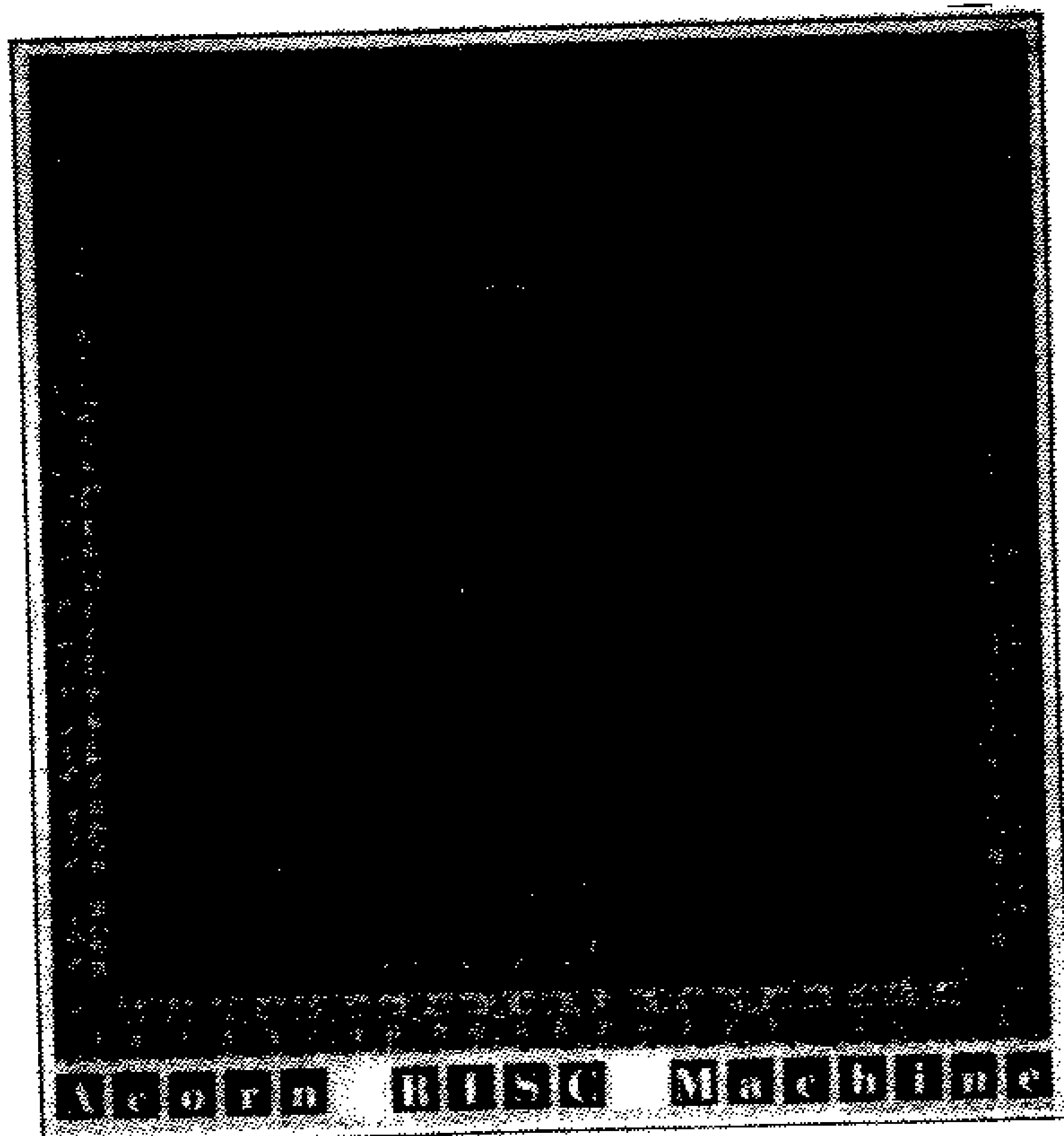
Tevens is gebruik gemaakt van het adres waar bijgehouden wordt waar de cursor in de regel staat, bij mij #E0 (het label col).

Om een tabel te vinden moeten de eerste twee bytes van de tabel #E3 en #C6 zijn (regel 13360 en 13490 terwijl de tabel afgesloten moet worden met #80 (regel 14010).
(in ieder geval worden de MINIAS tabellen herkent.)

Voor de CMOS monitor heb ik niets aangepast omdat ik deze niet heb en ook niet van plan ben aan teschaffen, daar ik bezig ben met een 65816 systeem waardoor ik de monitor beter hier voor aan kan passen.

Als er verder vragen opmerkingen of andere niet te stuiten opwellingen zijn laat dit dan zeker weten.

A.M. Vink
Lankheethoek 13
7546 BP Enschede
Tel. 053-775298



Inleiding
=========

Voor de mensen die een disc-drive bezitten en gebruik maken van de ED64-tekstverwerker en DATABASE V2.0, is dit programmaatje wellicht interessant. Het biedt de mogelijkheid om gegevens uit een bestand te verwerken in een textfile. Laat ik een voorbeeld noemen. Je wilt al je kennissen uitnodigen voor een knallende fuif. Met DATABASE V2.0 heb je een bestand opgebouwd waarin je de namen en adressen van deze kennissen hebt vastgelegd. Met de ED64-tekstverwerker schrijf je de uitnodiging. Vervolgens rolt voor alle kennissen een persoonlijk gerichte uitnodiging, met daarin verwerkt hun naam en adres, uit de printer.
Voor verschillende systemen zijn er al jaren programma's op de markt die deze manier van werken mogelijk maken. De broertjes van de Atom, Electron en BBC, maken hier geen uitzondering op.

Datafile
========

Zoals bekend, bestaat een bestand uit records. In dit geval is er voor elke kennis, een record. Zo'n record bestaat weer uit tien regels die we velden noemen. Het eerste veld wordt veld-0 genoemd, het volgende veld wordt veld-1 genoemd en ga zo maar door tot en met veld-9. Stel dat in veld-0 de achternaam, in veld-1 de voornaam, in veld-2 het adres, in veld-3 de postcode en de woonplaats en in veld-4 het telefoonnummer staat. Het is nodig om elk record op dezelfde manier in te delen zodat we de plaatsen weten van de verschillende gegevens. Dit bestand saven we onder de filename "data" op disc.

Textfile
========

In onze uitnodiging willen we gebruik maken van de gegevens die in de datafile te vinden zijn. Door in de tekst een apestaart gevolgd door een cijfer (0 tot en met 9) op te nemen zal nu het programmaatje "MIX" begrijpen dat hier het overeenkomende veld moet worden geprint. Voorbeeldje :

```
"Aan @1 @0"
"    @2"
"    @3"
"Hallo beste @1,"
```

Op deze manier wordt eerst een uitnodiging aan Jan gericht, dan aan Piet, evenveel uitnodigingen als je bestand records heeft. Weet wel dat je maximaal vierenzestig karakters per regel kwijt kunt en dat per veld maximaal dertig karakters van deze regel kunnen worden opgesnoept. Deze tekst saven we onder de filename "text" op dezelfde disc.

MIX
===

Om de tekst netjes te displayen heeft "MIX" wel steun nodig van "VDU6424". Ook deze moet op dezelfde disc gesaved zijn onder de naam "VDU6424". Dit doe je zo : "*SAVE VDU6424 9800 9EFE 9C04". Nu nog even het programmaatje "MIX" op dezelfde disc zodat je een vaste disc hebt waar je mee kunt werken zonder dat je "MIX" en "VDU6424" steeds bij elkaar hoeft te zoeken. Dit als volgt : "*SAVE MIX 2900 2BE0 CE86". En nu runnen maar met : "*RUN MIX". Behalve de eerste, verplichte ".!" (begin document) laat "MIX" het niet toe dat bewerkingscommando's worden gebruikt binnen de textfile. Daarentegen is het een eenvoudig programmaatje waarin ieder naar eigen inzicht veranderingen kan aanbrengen.

Toepassing
==========

Behalve voor het uitnodigen voor een fuif kun je andere, minder aangename doch nuttiger toepassingen bedenken. Denk maar eens aan het verzenden van nota's, circulaires en dergelijke. Zelf ben ik regelmatig in de weer met zulke dingen en ik heb (nu al) veel plezier aan dit programmaatje beleefd.

Guido de Roos
Sacramentsstraat 17
8911 GK Leeuwarden
telefoon 058-127150

```
 10 PROGRAM MIX (GdR-201086)
 20
 30 *NOMON
 40 *RUN VDU6424
 50 *LOAD data 2C00
 60 R=#2C00
 70 DO
 80  R=R+1
 90 UNTIL ?(R-1)=255
100 WHILE ?R<>254
110  R=R+1
120  *LOAD text 2C00
130  T=#2C00
140  WHILE ?T<>13
150   T=T+1
160  WEND
170  PRINT $12"PAPER"$7
180  INKEY K
190  PRINT $13"       "$13$2
200  WHILE ?T<>4
210   WHILE ?T=CH"@"
220    N=(T?1)-48
230    T=T+2
240    *LOAD data 2C00
250    D=R
```

```
260      F=0
270      WHILE F<>N
280       DO
290        D=D+1
300       UNTIL ?(D-1)=13
310       F=F+1
320      WEND
330      PRINT $D
340      *LOAD tekt 2C00
350     WEND
360     XIF ?T=13
370      PRINT '$21
380      HTAB 8
390      PRINT $6
400     ELSE PRINT $(?T)
410     T=T+1
420    WEND
430    *LOAD data 2C00
440    DO
450     R=R+1
460    UNTIL ?(R-1)=255
470    PRINT ''''$3
480 WEND
490 PRINT "READY"'
500 END
```

Verklaring
==========

```
 60 t/m  90 : Zoek het eerste record.
              (Negeer het mainframe van het bestand.)
100         : Is het eind van de datafile bereikt ?
110         : Negeer het record-nummer.
130 t/m 160 : Zoek het eerste karakter van de textfile.
              (Na de eerste carriage-return met ervoor ".i".)
170 t/m 190 : Voer papier in en start de printer.
200         : Is het eind van de tektfile bereikt ?
210         : Moet er een veld worden geprint ?
220  &  230 : Zo ja, welk veld ?
250 t/m 330 : Print het betreffende veld.
360         : Moet er een linefeed worden geprint ?
370 t/m 390 : Zo ja, print een linefeed en spring in.
400         : Zo nee, print het karakter.
410         : Naar het volgende karakter.
440 t/m 460 : Naar het volgende record.
470         : Print vier linefeeds en stop de printer.
```

Archimedes: de snelste PC ter wereld

De Engelse computerfabrikant Acorn Computers Ltd. heeft een paar weken geleden een geheel nieuw computersysteem geïntroduceerd. Nu al is vrijwel iedereen het er over eens dat deze nieuwe serie van computers tot de meest revolutionaire ontwerpen behoort en dat een relatief kleine fabrikant zich in een keer aan de top van de computerindustrie heeft geplaatst. De kern van de vier nieuwe machines vormt een door Acorn zelf ontworpen 32-bits RISC-processor die luistert naar de naam ARM (Acorn Risc Machine).

In het begin van de jaren tachtig is Acorn op de computermarkt verschenen met een voor die tijd zeer modern computersysteem dat luisterde naar de naam Acorn Atom. De computer was helemaal opgebouwd met gewone chips, dus geen komplexe custom-design-komponenten. Na de Atom is alles snel gegaan, Acorn verwierf de opdracht om speciaal voor de BBC een computersysteem te ontwerpen, naderhand bekend geworden als respektievelijk de BBC-A, BBC-B, BBC-B+ en Acorn Master. Een ding hebben al deze computers gemeen; ze ontlenen hun kracht aan de 6502 en ze zijn erg flexibel en snel. In tijden dat veel fabrikanten de sprong voorwaarts maakten met 16-bit-processoren zoals de 68000, bleef Acorn het 8-bit-systeem trouw. Tijdens de stormachtige jaren die volgden op de introduktie van de BBC-serie computers, moest de Italiaanse computerfabrikant Olivetti er aan te pas komen om ervoor te zorgen dat Acorn niet failliet ging en voorgoed van het firmament zou verdwijnen. Deze malaise bleek vooral veroorzaakt te worden door een te zwak management. Gedurende deze turbulente tijd bleef echter steeds een groepje technici in de Verenigde Staten koortsachtig werken aan het ontwerp van de RISC-processor. Medio 1986 kwamen de eerste prototypen hiervan ter beschikking en had Acorn een voorsprong op konkurrerende fabrikanten waaronder zelfs IBM. Tijdens de HCC-dagen van 1986 was er al een werkende Acorn Risc Machine (ARM) te zien als co-processor van een BBC. Het leuke van deze RISC-machine was dat alle software die in BASIC op de BBC geschreven was, probleemloos liep op de ARM, alleen met een veel hogere snelheid (tot zo'n tien keer sneller). Nu, anno 1987, is de definitieve serie personal computers geïntroduceerd en kan vanaf september de grootschalige verkoop van het computersysteem "Archimedes" verwacht worden. Tot die tijd worden slechts kleine aantallen uitgeleverd omdat de echte produktie nog niet opgestart is.

Wat maakt een RISC-processor zo bijzonder?

Bij de ontwikkeling van moderne micro-processors is een lijn te zien van steeds krachtiger wordende instrukties. Ook het aantal instrukties neemt zienderogen toe. Voor de programmeur betekent dit dat hij een grote keus heeft uit instrukties en dat zelfs vermenigvuldigen en delen in machinetaal zonder komplexe subroutines mogelijk is. In gewone processoren zijn al deze instrukties opgebouwd uit een aantal kleinere instrukties, die men ook wel mikro-kodes noemt. Eigenlijk zijn de instrukties van bijvoorbeeld een 68020 kleine subroutines. Het nadeel van deze ontwikkeling is dat het uitvoeren van deze instrukties door deze komplexe opzet relatief traag wordt. Hieraan kan wat gedaan worden door gebruik te maken van een hogere klokfrekwentie en bijvoorbeeld een 16-bit-databus. Bij de RISC-processor (Reduced Instructions Set Computer) wordt een andere weg bewandeld. Een RISC-processor bevat relatief weinig instrukties die op snelheid ontworpen zijn. Dat lagere aantal instrukties hoeft geen al te groot probleem te zijn, want uit de statistiek blijkt ook nu weer dat de 80/20-regel ook bij software-schrijvers bestaat. 20% van de instruktie zorgt voor 80% van het programma, oftewel 80% van de instrukties wordt bijna nooit gebruikt. De hoge snelheid wordt bereikt door in een RISC-processor de instrukties via hardware uit te voeren. Bovendien wordt er door de struktuur die in de instrukties zit voor gezorgd dat er in de software nagenoeg geen gebruik gemaakt hoeft te worden van jumps en subroutines. Bij de ARM is de opkode 32 bit breed. Van deze 32 bits zijn er vier gereserveerd voor kondities, zodat men alle instrukties konditioneel kan aanroepen. Vandaar dat zo'n processor erg veel registers bezit; de ARM heeft maar liefst vijfentwintig 32-bit-registers. Dankzij dit grote aantal registers en de speciale architektuur kunnen vele opdrachten parallel verwerkt worden. Verder wordt er in technologie veel gebruik gemaakt van pipe-lining. Pipelining is een techniek die afkomstig is uit de main-frames waarbij tegelijk aan verschillende instrukties gewerkt wordt. Terwijl een instruktie wordt uitgevoerd, wordt de volgende alvast gedekodeerd en de daaropvolgende al uit het geheugen gehaald. In de ARM worden de instrukties in maximaal twee klokcycli verwerkt. Dit betekent dat bij de huidige klokfrekwentie van 8 MHz vier miljoen instrukties per sekonde (4 MIPS) verwerkt kunnen worden. De Acorn Risc Machine is de eerste RISC-processor die kommercieel gebruikt gaat worden. RISC-processoren zullen in de praktijk meestal gebruikt worden in kombinatie met compilers om op deze manier toch gebruiksvriendelijk te zijn voor de programmeur. Voor de ARM zijn op dit moment al verschillende compilers beschikbaar.

De processor is opgebouwd in

de 2-μm-technologie en heeft een oppervlakte van 50 mm². Verder heeft hij een 32-bit-databus; een 26-bit-adresbus kan daarmee maar liefst 64 Mbyte aan geheugen adresseren. Tenslotte is het mogelijk om met virtueel geheugen te werken. Speciaal voor real-time-toepassingen is er een zeer snelle response op interrupts (gewoonlijk binnen 2 mikrosekonden en maximaal binnen 6 mikrosekonden). Slechts één voedingsspanning van 5 volt is nodig om deze CMOS-chip met een opgenomen vermogen van 100 milliwatt te voeden.

Voor bezitters van PC's is er het Acorn Springboard, een kaart die in een van de uitbreidingsslots van de PC past. Met dit board is het mogelijk dat ook een gewone PC gebruik kan maken van de hoge snelheid die bij RISC-technologie hoort. Op het board treft men naast de met 8 MHz geklokte RISC-processor 1 of 4 Mbyte aan geheugen aan. Dankzij dit board krijgt de PC de mogelijkheid om twee grote programma's gelijktijdig uit te voeren. Terwijl de PC als tekstverwerker gebruikt wordt, kan het Springboard bijvoorbeeld data verwerken van een meting die op dat moment wordt uitgevoerd. In totaal kunnen vier Springboards in een PC geplaatst worden. Voor deze Springboards is momenteel naast een assembler en debugger ook C, Prolog, BASIC, Lisp en Fortran beschikbaar. De benchmarks van de Archimedes tonen duidelijk aan dat het bij deze techniek niet om een kleine procentuele vooruitgang gaat, maar de toename in verwerkingssnelheid bijna ongelooflijk is. Zelfs een Compaq 386, nog niet zo lang geleden als een héél snel systeem op de markt gebracht, heeft geen antwoord op de snelheid van dit Engelse wonderkind.

RISC-architektuur staat ook nog een heel andere mogelijkheid van programmeren en werken met een computer toe. De taal van RISC-processoren, machinetaal dus, hoeft niet de taal te zijn waar de eindgebruiker van de computer mee te maken heeft. Bij Archimedes worden software-emulatoren geleverd die het mogelijk maken de ARM te programmeren alsof het een 8088 danwel een 6502 is. Het is

New Personal Computer World magazine benchmarks for interpreted BASIC

Machine	Language	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
300 Series:	BBC Basic	0.26	0.28	1.02	42	
RM VX 386	GW Basic	0.89	1.05	8.09	35.9	
IBM Model 30	Basic	2.6	3.4	25.4	[illegible]	
IBM Model 50	Basic	1.4	2.04	12.5	[illegible]	
IBM PC	BasicA	6.2	8.2	47.0	[illegible]	
Amiga 2000	AmigaBasic	1.7	2.7	6.7	[illegible]	
Olivetti M28	Basic	2.1	2.0	15.0	33.6	
Atari ST	ST Basic	1.5	3.5	7.9	44.8	
Master 128	BBC Basic	2.5	4.3	43.0	14.2	
Compaq 386	GW Basic	1.0	0.96	3.85	25.5	

Notes: 300 Series figures are Acorn measurements of BBC BASIC in RAM using PCW algorithms. All other figures are taken from PCW reports.

*Not available for floppy drives.

Sieve of Eratosphenes for interpreted BASIC

Machine	Language	Speed
300 Series:	BBC Basic	8.4
Compaq 386	Compaq Basic	21
Atari 1040ST	ST Basic	85
Amiga 1000	AmigaBasic	66
IBM PC AT 8 MHz	GW Basic	61

Note: 300 Series figures are Acorn measurements of BBC BASIC in RAM using Byte Magazine programs. The program does one iteration of prime numbers up to 7000. All other figures are taken from Byte reports.

Compiler Tests

Dhrystone (version 1.1 test). (C to proposed ANSI standard)	300 Series	4560 Sec

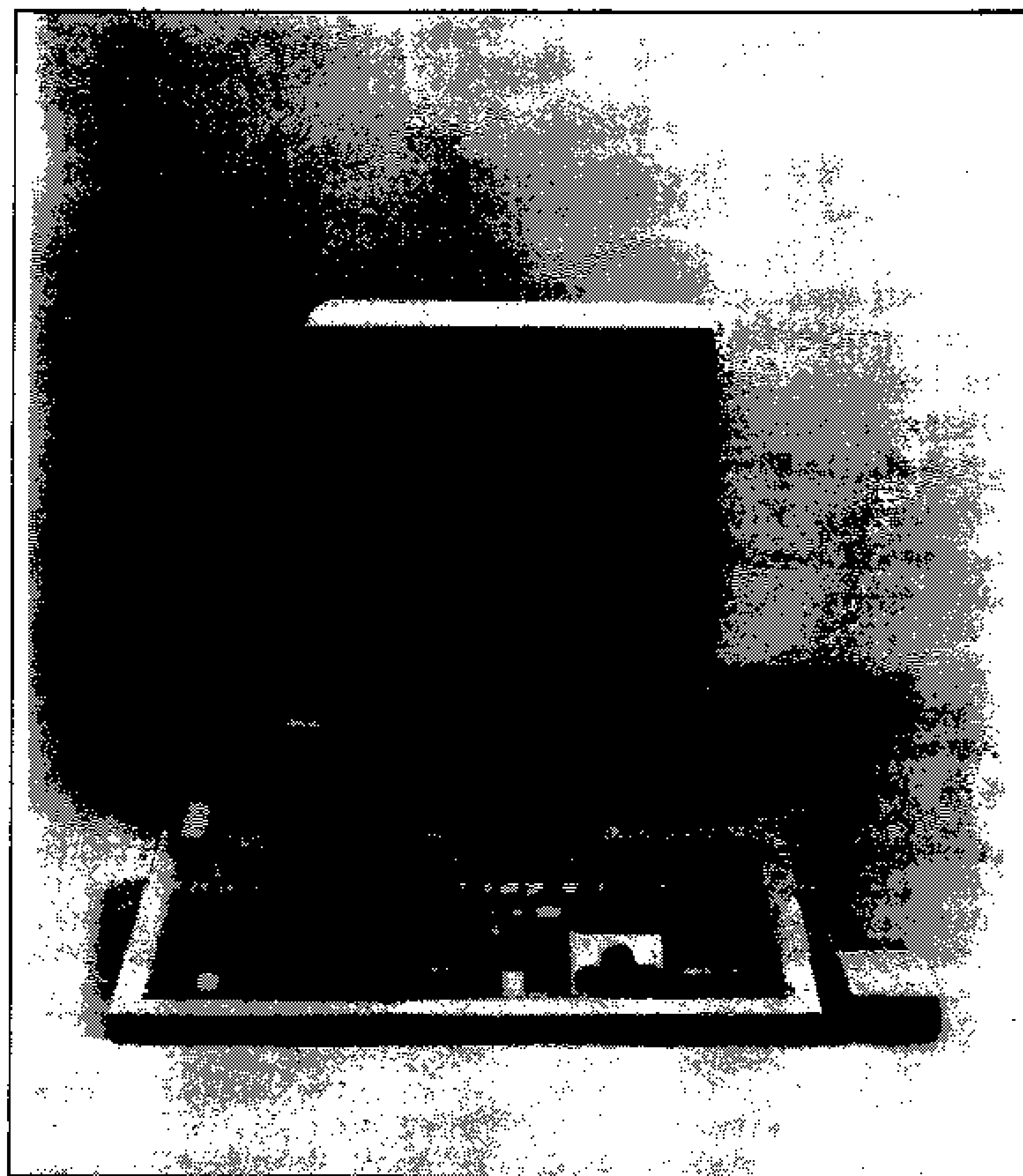
'The ARCHIMEDES High Performance Computer Systems, the fastest in their class to date.

dus mogelijk dat op een systeem waarin een RISC-processor gebruikt is, machinetaal loopt die voor een heel andere microprocessor geschreven is. Sterker nog, op de Archimedes loopt machinetaal voor de 6502 aanzienlijk sneller dan op de toch beslist niet trage BBC. In de praktijk zal er echter weinig noodzaak zijn om in machinetaal te programmeren, omdat de BASIC die bij de RISC-processor hoort, BBC-BASIC-V, qua snelheid meer lijkt op machinetaal dan op BASIC. Wel is het mogelijk een computer te ontwerpen die zich als een kameleon gedraagt. Voor de ene toepassing is het een C64, voor de andere een Atari ST en voor de derde bijvoorbeeld een Commodore Amiga of een IBM PC. Daardoor is een zeer universeel systeem ontstaan waarop bijna alle software te gebruiken is.

Archimedes, een systeem dat toekomst heeft

RISC-processoren kunnen dan nog zo snel zijn, aan een processor alleen heeft de gebruiker niets. Vandaar dat men in Engeland met spanning zat te wachten op de introduktie van de eerste RISC-computer. Half juni was het dan zover: Archimedes werd geïntroduceerd. Voor de ongelooflijk lage prijs van achthonderd Engelse ponden is de A300 te koop, kompleet met 512 Kbyte aan geheugen en een 800 Kbyte 3½"-floppy-disk-drive. Het is mogelijk om het systeem naderhand nog verder uit te bouwen met een losse upgrade-kit die de ontbrekende komponenten van het moederbord bevat. In het begin zijn dus bij de eenvoudigste versie van Archimedes een aantal plaatsen op het moederbord leeg. Verder

kan in de behuizing naar keuze een tweede floppy-disk of een hard-disk (20 Mbyte) ingebouwd worden. De computer wordt kompleet geleverd met software, waaronder BBC-BASIC-V, een 6502-emulator en een muis. Het is goed mogelijk dat in Nederland ook nog de 8088-emulator standaard wordt meegeleverd, waardoor de computer ook alle MS-DOS software kan verwerken. Om ook op hardware-nivo bestand te zijn tegen beveiligingen in de software komt er een 80186 co-processor ter beschikking samen met de originele in PC's gebruikte floppy-controller. Op dit uitbreidingsbord kan een extra 5¼"-disk-drive worden aangesloten zodat de originele beveiligde schijf zonder problemen verwerkt kan worden.



De sterkste kant van het systeem is zonder enige twijfel het grafische display. De computer heeft een pallet met 4096 verschillende kleuren waaruit er steeds 256 gekozen kunnen worden. Zelfs bij de allerhoogste resolutie van 640 x 512 punten heeft men de keuze uit 16 verschillende kleuren. Voor deze modes is wel een speciale monitor zoals de NEC multisync noodzakelijk. In totaal zijn er 21 verschillende schermmodes aanwezig, waaronder de bekende teletekst-mode. De mogelijkheden met geluid blijken ook echt zeer bijzonder te zijn. In totaal 8 stereo-kanalen zijn in de computer aanwezig en er kan optioneel een MIDI-interface aangeschaft worden.

Het toetsenbord heeft de bekende IBM-layout gekregen, net zoals de konnektoren van de interfaces aan de achterkant. Overigens zal het toetsenbord minder gebruikt worden omdat ook bij deze computer icons en windows in het operating system ARTHUR zijn opgenomen. Met de muis kan het hele systemen bestuurd worden.

Zoals het er nu naar uitziet heeft Acorn met de introduktie van deze nieuwe serie van personal computers een zeer scherp geprijsd en ijzersterk systeem op de markt gebracht. Omdat de computer kompatibel is met zijn voorgangers (voornamelijk dankzij de 6502-emulator en de zeer goede systeem-software) is er al direkt bij de introduktie een enorme hoeveelheid goede software beschikbaar die erg snel verwerkt kan worden. Verder is er de mogelijkheid om MS-DOS op alle systemen te gebruiken in kombinatie met zeer goede graphics. Dit lijken allemaal punten die het sukses van dit systeem moeten garanderen, maar zoals zo vaak is het de konsument die bepaalt of het systeem wel of niet suksesvol is.

(EA-365)

DISC MENU GUIDO DE ROOS

Inleiding

Een eigenschap van GDOS is dat men filenames van maar liefst 15 karakters kan gebruiken. Een twede eigenschap is dat per disc 78 files kunnen worden opgeslagen. Helaas heeft dit tot gevolg dat bij een *CAT lang niet alle files gedisplayed worden op ons beperkte Atom-scherm. Om een file in te laden zal men met GDOS vaak meer typewerk moeten verrichten dan met het originele DOS het geval is. Om juist dit te ondervangen heb ik het DISC MENU geschreven. Zelf heb ik dit programma vanaf #0400 geinstalleerd om deze permanent te kunnen aanspreken. Wanneer dit programma zich buiten het gebied #1000 - #A000 bevindt, is het nodig om P-Charme aan te passen volgens A.N. 5-3, pagina 8. Bovendien is het noodzakelijk om een schakelkaart met schakelsoft te hebben. Vervolgens maakt DISC MENU gebruik van enkele statements, die niet direkt standaard of algemeen zijn te noemen. Later zal ik hier nog op terug komen. Mijns inziens weegt het bedieningsgemak ruimschoots op tegen de moeite van het installeren van de ondersteunende statements.

Werking

Met "DISC" wordt het programma aangeroepen en zal netjes in 3 kolommen de eerste 39 files displayen (indien aanwezig). Voor elke file laat het de eerste 7 karakters van de filename zien. Ervoor staat het filenummer. Bovenaan het scherm vindt men de eerste 7 karakters van de titel van de disc en het aantal files wat op de disc staat. Door het filenummer in te typen (wat voor de filename staat), kiest men welke file ingeladen wordt. Als dit nummer met een nul begint, moet deze ook worden ingetypt. Achter "CHOOSE:" verschijnt nu het ingetypte nummer en de desbetreffende file wordt ingeladen. Wanneer een file niet bij de eerste 39 staat, moet een carriage-return worden gegeven zodat de volgende 39 files worden gedisplayed (indien aanwezig). Dit is ook van toepassing als een file op een andere disc staat. Nu kan men opnieuw zijn keuze bepalen. Als de file ingeladen is, wordt het filenummer voor de file geinverteerd. Hieraan kan men zien welke file ingeladen werd. Rechts-boven op het scherm, op de plaats van "CHOOSE:??", wordt de vraag "EXECUTE:?" gesteld. Deze kan worden beantwoord met "Y", "N" of een carriage-return. Verklaring :

- Y (Yes)

  Er wordt uitgezocht of de file BASIC of machinetaal is. Als het BASIC is, wordt het gerund en als het machinetaal is, wordt er gelinkt naar het executieadres.

- N (No)

  Als de file BASIC is, wordt naar de textpage gesprongen

waar deze file staat. Als het machinetaal is, wordt naar textpage #29 gesprongen.

- carriage-return (Return to DISC MENU)

  De eerste 39 files worden weer gedisplayed zodat opnieuw kan worden gekozen. Zo is het mogelijk meerdere programma's in te laden die samen werken of op verschillende discs staan.

Als er een graphic wordt ingeladen, met het startadres #8000, wordt deze direkt in de hoogste grafische mode gedisplayed en zal er worden gewacht tot er een carriage-return gegeven wordt voordat men weer kan kiezen uit de eerste 39 files.

Support

Zoals ik eerder vermeldde, heeft DISC MENU ondersteuning nodig. Het programma draait op P-Charme en maakt daarnaast gebruik van andere statements. Te weten :

- FILL

  Deze komt voor in de GAGS-ROM, een utility die beslist op de schakelkaart thuis hoort.
  (A.N. 4-6, pagina 31.)

De volgende statements heb in mijn schakelsoft opgenomen. Ook is het mogelijk deze als P-Charme-uitbreiding te installeren.

- CURSOR

  A.N. 4-2, pagina 74.

- CON en COFF

```
 10 PROGRAM CON/COFF
 20
 30 DIMLL2;F.N=0 T.2;LLN=-1;N.;P.$21;P=A;GOS.a;P.$6;P=A;GOS.a
 40 $T="CON";T=T+L.T;?T=LL0/256|#80;T?1=LL0%256;T=T+2
 50 $T="COFF";T=T+L.T;?T=LL1/256|#80;T?1=LL1%256;T=T+2
 60 ?T=#80;A=P;T!1=A;E.
 70
 80a[:LL0;JSR#C4E4;LDA@#80;JMPLL2
 90:LL1;JSR#C4E4;LDA@#00
100:LL2;STA#E1;JMP#C55B;]R.
```

- CLR

  A.N. 3-2, pagina 50. Verander de naam "CLS" in "CLR", anders komt dit statement niet aan bod. Er bestaat namelijk al een andere "CLS". Deze is opgenomen in de GAGS-ROM en verschilt met de hier genoemde.

- CPC

  Dit is een verbeterde versie van de eerder verschenen "CPC",

(A.N. 4-7, pagina 31.)

```
 10 PROGRAM CPC
 20 DIMLL2;F.N=0 T.2;LLN=-1;N.;P.$21;P=A;GOS.a;P.$6;P=A;GOS.a
 30 $T="CPC";T=T+L.T;?T=LL0/256|#80;T?1=LL0%256;T=T+2
 40 ?T=#80;A=P;T!1=A;E.
 50
 60a[:LL0;JSR#C4E4;CLC;LDA#DE;ADC#E0;TAY;LDA@#20
 70LDX#DF;CPX@#80;BNELL2
 80:LL1;STA#8000,Y;INY;CPY@#00;BNELL1
 90:LL2;STA#8100,Y;INY;CPY@#00;BNELL2
100JMP#C55B;]R.
```

- RESET

Naar A.G. Geraets, A.N. 4-3, pagina 75.

```
 10 PROGRAM RESET
 20 DIMLL2;F.I=0 T.2;LLI=-1;N.;P.$21;P=A;GOS.a;P.$6;P=A;GOS.a
 30 $T="RESET";T=T+L.T;?T=LL0/256|#80;T?1=LL0%256;T=T+2
 40 ?T=#80;A=P;T!1=A;E.
 50
 60a[:LL0;JSR#C4E4;LDA#12;STA#0E;LDY@#00;STY#0D;DEY
 70:LL1;INY;LDA(#0D),Y;CMP@#0D;BNELL1
 80JSR#CDBC;LDA(#0D),Y;BMILL2
 90INY;BNELL1
100:LL2;INY;JSR#CDBC;LDA#0D;STA#23;LDA#0E;STA#24;JMP#C55B;]R.
```

Listing
=======

```
 10 PROGRAM DISC MENU
 20 REM GDR-SOFT 1986
 30
 40 PROC FILE
 50  IF I<10 PRINT Z
 60  PRINT I,":"
 70  FOR L=Z TO 6
 80   PRINT $F?(((I-1)*#10)+L)
 90  NEXT L
100  XIF I%3
110   HTAB ((I%3)*11)
120   G=G+11
130  ELSE G=G+10
140 PEND
150
160 PROC CHOOSE
170  C=Z
180  DO
190   INKEY K
200  UNTIL K>#2F AND K<#3A OR K=#0D
210  IF K=#0D GOTO 320
220  C=K-#30
230  CURSOR 31,Z
240  PRINT C
250  DO
260   INKEY K
```

```
270  UNTIL K>#2F AND K<#3A
280  C=(C*10)+K-#30
290  CURSOR 30,Z
300  IF C<10 PRINT Z
310  PRINT C
320 PEND
330
340 REM MAIN PROGRAM
350 *DOS
360 *DIR
370 @=0
380 Z=0
390 COFF
400 CLR
410 F=#2010
420 N=?#200F
430 PRINT "TITLE:"
440 FOR I=Z TO 6
450  PRINT $#2000?I
460 NEXT I
470 PRINT " FILES:"
480 IF N<10 PRINT Z
490 PRINT N," CHOOSE:??"'
500 FILL #8020,#8040,#CC
510 G=#8040
520 I=1
530 WHILE I<=N AND I<40
540  FILE
550  I=I+1
560 WEND
570 IF (I-1)%3 G=G+(#20-(G%#20))
580 FILL G,(G+#20),#CC
590 CHOOSE
600 G=#8040
610 WHILE C=Z OR C>I AND I<=N
620  CURSOR 30,Z
630  PRINT "??"
640  VTAB 2
650  CPC
660  WHILE I<=N
670   FILE
680   I=I+1
690  WEND
700  IF (I-1)%3 G=G+(#20-(G%#20))
710  FILL G,(G+#20),#CC
720  CHOOSE
730 WEND
740 IF C=Z OR C>N RUN
750 G=#8040+(((C-1)%39)/3*#20)+((((C-1)%39)%3)*11)
760 ?G=(?G):#80
770 G?1=(G?1):#80
780 F=F+((C-1)*#10)
790 F?#0F=#0D
800 S=(#2500!((C-1)*#08))&#FFFF
810 E=(#2502!((C-1)*#08))&#FFFF
820 IF S=#8000 CLEAR 4
830 *NOMON
840 *OSCLI "LOAD ",$F," ",&S,$#0D
```

```
 850 WHILE S=#8000
 860  DO
 870   INKEY K
 880  UNTIL K=#0D
 890  RUN
 900 WEND
 910 CURSOR 23,Z
 920 PRINT "EXECUTE:?"
 930 WHILE K<>CH"Y" AND K<>CH"N" AND K<>#0D
 940  INKEY K
 950 WEND
 960 CURSOR 31,Z
 970 PRINT $K
 980 IF K=#0D RUN
 990 *MON
1000 @=8
1010 CON
1020 CLR
1030 ?#12=#29
1040 XIF E=#C2B2 OR E=#CE86 OR (E=S AND ?E=#0D)
1050  E=Z
1060  ?#12=S/#100
1070  RESET
1080 ELSE IF E>#27FF AND E<#A000 AND K=CH"Y" LINK E
1090 IF K=CH"Y" AND E=Z RUN
1100 END
```

Wie vragen of opmerkingen heeft, kan altijd bij mij terecht.

Guido de Roos
Sacramentsstraat 17
8911 GK Leeuwarden
telefoon 058-127150

P.S. Wie schrijft de software waarmee files naar het originele DOS kunnen worden geschreven ?

STORINGEN en hun oorsprong.
===========================

Voor u gelezen in: "Oak Leaves" (Uitgave van de Canadese Atomclub "ATOMOTA" een artikel geschreven door Rick Bales.) vrij vertaald door: J.Ballijns.

De meesten van ons hebben onze ATOM's al een hele tijd. In computer termen is een 8-bit machine oude technologie. Met andere woorden Atom's zijn oude machines. En oude machines willen nog wel eens kuren vertonen of er mee ophouden. Vandaar dat we eens moeten praten over wat te doen om deze storingen op te lossen.

Het feit dat de Atom oude technologie is heeft beide voor- en nadelen. Een nadeel is dat het soms moeilijk is om een component te vinden, omrede dat deze niet meer gangbaar zijn of reeds uit de produktie zijn genomen. Dan is de enige mogelijkheid nog oude voorraad of afgedankte machines.

Een voordeel is dat oude technologie vaak goedkoop is. Vele van ons betaalde vijf honderd dollar voor een kale atom terug in die dagen dat het enige alternatief in de zelfde prijs klasse de Vic 20 was. Nu kun je een 6502 cpu kopen voor onder de 3 dollar en 2114 Ram chips voor 50 cents. De meeste reparaties aan de Atom zijn dus relatief goedkoop. Het opsporen en vinden van de oorzaak van de storing is echter voor velen een probleem waar we wat aan willen doen.

Ten eerste, laten we eens overwegen wat er verkeerd kan gaan. In een woord: Alles. Maar dat, wat waarschijnlijk weigerd, of kapot gaat, is veel meer beperkt; zolang we praten over normaal gebruik in tegen stelling met bruut geweld. Het printboard zal normaal geen storing geven. Het zijn hoofdzakelijk de componenten die er op gemonteerd zijn die storing kunnen geven en deze kunnen allen worden vervangen. Sommigen, zoals IC's (chips), zijn gemakkelijk te vervangen, daar zij, in tegenstelling met vele andere micro's, alle chip's in de Atom in voetjes zijn gemonteerd. Het is dus erg makkelijk om een chip te verwijderen en te vervangen door een nieuwe. Echter niet elke chip die niet, of niet goed functioneerd moet vervangen worden door een nieuwe, straks meer hierover. Andere onderdelen zoals condensatoren, weerstanden, crystallen en IC voetjes zijn op het printboard vast gesoldeerd en daarom moeilijker te vervangen. Een onderdeel is erg moeilijk om aan te werken nl. het toetsenbord (keyboard). Gelukkig is het toetsenbord erg solide en behoefd zelden te worden vervangen. Mijn eigen toetsen bord is zoveel gebruikt dat vele toetsen piepen en glimmen van slijttage maar tot op heden hebben ze nooit gefaald.

Oke, laten we zeggen uw Atom heeft de geest gegeven.
U eerste overweging moet dan zijn het opsporen van het probleem. Wat u nodig heeft voor het diagnostiseren van het probleem is:

1) Een oscilloscoop. 2) Een digitale logic probe. 3) Een multimeter. 4) Een set schroevendraaiers incl. Phillips types.
En voor reparatie heeft u nodig 5) Een laag-wattage soldeerbout met een smalle tip. 6) Een desoldeertool en wat desoldeer wick.

Nu de meesten van ons hebben geen scope en vele hebben geen logic probe. Waarschijnlijk hebben sommige alleen de vereiste schroevendraaiers en dan nog niet de vereiste phillips types. Wanhoop dan niet. Ik heb geen scope en ben de tel kwijt van het aantal Atoms die ik heb gerepareerd. Ik heb wel een logic probe maar ik heb het nog niet gebruikt (ben de handleiding kwijt). Ik heb wel een multimeter en die gebruik ik veel. En ik kan dikwijls mijn phillips schroevendraaiers niet vinden en gebruik dan een smalle gewone.

Als je geen multimeter hebt, dan raad ik je sterk aan er een aan te schaffen en leer er mee om te gaan. Het mag best een goedkoop apparaatje wezen, U zal zien hoeveel plezier u er van beleeft. De soldeerbout en de-soldeer spullen heeft u alleen nodig als de reparatie dit vereist. Maar het is het waard om deze spullen te hebben, daar we er vroeg of later toch wel gebruik van zullen maken. Vergeet de oscilloscope geheel en al tenzij u niet weet wat te doen met uw geld. En wat de logic probe aangaat koop er een u kunt hem altijd gebruiken als swizzle stick op een cocktail party.

Terug naar waar we mee bezig waren. Het eerste wat we moeten doen is het controleren van onze stroombron. Is er voltage op de uitgaande lijnen?. En is de voltage dicht bij de correcte waarde? Als de regulators op de print niet gebruikt worden dan moeten de links 6 en 7 zijn aangebracht zo als beschreven in het technisch manuel pag. 13 .

Vijf volt op de uitgang van onze krachtbron is geen waarborg dat er voldoende voltage verder op de print aanwezig is. Het is belangrijk om te controleren of er voldoende spanning op het midden van het printboard is, daar er toch wel wat spanning verlies optreed. De meeste componenten hebben 5 volt plus of minus een kwart volt nodig. Plaats de positive pen van de multimeter op pin 1 van de CPU (6502) en de andere pen op pin 8 en check voor een aflezing van 4.75 en 5.25 volts. Als het buiten dit raam valt dan moet de krachtbron aangepast worden.

Het gebruik van een goede geregulieerde 5 volts krachtbron, sterk genoeg om al u uitbreidingen ook van stroom te voor zien, wordt dringend aanbevolen. Denk om de links 6 & 7 als je nu pas overgaat op een apparte 5 volt powersupply.

Indien u probleem door voorgaande procedures niet is opgelost dan is het mischien raadsaam om het geheugen te testen. Doe een Ram test met behulp van een van de daarvoor geschreven programma's. Controleer alle Ram in uw systeem uitgezonderd, dat in page zero, dat in het video gedeelte en dat gedeelte waarin het test programma staat. (running Ramtest in page zero laat je machine hangen). Indien er fouten optreden lokaliseer de betreffende chips en vervang ze. Hier volgt een lijst van de chips met de daarbij behorende adressen:

IC nummer	ADRESSEN		
10 & 11	#2800	-	#2BFF
12 & 13	#2C00	-	#2FFF
14 & 15	#3000	-	#33FF
16 & 17	#3400	-	#37FF
18 & 19	#3800	-	#3BFF

Indien u een 64K kaart of de 16/32K of een geheugenkaart heeft moet u zelf maar uitzoeken welke chip welk geheugen bestrijkt.

Om page zero Ram te controleren, verwijder de IC's 51 en 52 en zet ze even apart. Verwijder de IC's 10 en 11 en plaats ze in de voetjes van de IC's 51 en 52. plaats nu de apart gezette IC's in de voetjes van de IC's 10 en 11. Nu nogmaals de test runnen voor het adres gedeelte #2800 - #2BFF. Nadat de test is afgewerkt hoeven de IC's niet meer te worden verwisseld.

Indien u een of meer fouten Ram lokaties ontdekt in dit testing proces, is het soms niet noodzakelijk om te gaan zoeken naar nieuwe IC's. Het probleem kan veroorzaakt worden door wat oxidatie op de pootjes van een of meer IC's. Dit is vaak te verhelpendoor de IC's wat te wiebelen in de voetjes,dus eerst verwijderen uit het voetje en dan weer terug plaatsen. Dit is vaak genoeg om de oxidatie af te schrapen en weer goed contact te maken tussen chip en voetje. In feite is deze procedure zo simpel dat het mischien een goed idee is om dit als routine te doen, of als een soort preventief onderhoud.

Video Ram is het makkelijkst te testen omrede de fouten ommiddelijk zichtbaar gemaakt kunnen worden. Indien uw tekst beeld geen rare karakters vertoond nadat u op BREAK heeft gedrukt dan zijn de IC's 32 en 33 op z'n minst gedeeltelijk goed. Voer het volgende programma in en run het:

```
10 CLEAR 4
20 GOTO 20
```

U moet nu een totaal blank mode 4 graphic beeld zien (als u alle video ram heeft gemonteerd. Laat de Atom zo zitten voor een uur en controleer of het scherm blank blijft. Zo niet, merk die plaatsen met een viltstift op het scherm. Stroom af. Verwijder IC's 34 en 35 nu stroom weer aan. Run het programma opnieuw. Door het verwijderen van genoemde IC's ziet u nu een groot wit blok op u scherm over de gehele breedte. Als dit blok enige van de gemerkte punten op het scherm bedekt, dan heeft u de foute IC gelokaliseerd. Is dit niet het geval dan stroom weer af en herplaats de IC's weer in hun voetjes. Verwijder nu het volgende paar IC's nl. 36 & 37. Run het programma weer en continueer op dezelfde manier totdat u alle foute video ram heeft gelokaliseerd. Merk op: dat ik nog niet de IC's 32 & 33 heb behandeld. Deze zijn nodig voor het normale tekst beeld zodat als we deze chips verwijderen dan is het gehele beeld blank als de Atom aan staat. Om deze te testen gebruiken we dezelfde procedure als bij het testen van page zero RAM. Verwissel de IC's 32 & 33 met een paar andere IC's in de video sectie en run het bovenstaande programma opnieuw. En dat draagt dus zorg voor het video Ram gedeelte.

Terwijl we met het video gedeelte van de ATOM bezig zijn wil ik u nog even wijzen op een paar meest gewone voorkomende oorzaken van problemen die ik gevonden heb in weigerende machines. Een probleem met de Video Display Generator chip (IC 31) kan de oorzaak zijn dat het gehele systeem hangt. Ongeveer 80 procent van de weigerende ATOM's waar ik mee te maken heb gehad waren problemen veroorzaakt door dit IC of het daarbij behorende circuit. De 6847 VDG chip welke het beeld verzorgd is een robuste chip (ik ben er nog maar een tegengekomen die fout was). Maar vele ATOM eigenaren hebben dingetjes toegevoegd aan deze sectie en dit verhoogd de kans op problemen met deze chip. Deze toevoegingen zijn o.a.: de Colour kaart en de Screen Noise Eliminator Board. Speciaal deze kaarten vereisen verwijdering van de 6847 en deze plaatsen op het printplaatje wat weergeplaatst wordt in het voetje van de 6847. Deze kaarten hebben soms pennen die niet goed passen in het voetje (zijn te dik) het resultaat is een slecht contact in het IC voetje. Het lijkt mischien raar dat een zware passing in het voetje de oorzaak kan zijn van een slecht contact maar ik verzeker u dat deze zware passingen vaak de oorzaak is van problemen als vervorming van het voetje en wat al niet meer resulterende in slechte contacten etc. 80 procent vav de slecht of helemaal niet werkende ATOMS had als oorzaak slechte contacten.

Hoe dit probleem te lijf te gaan vraagt wat meer werk. De beste oplossing is het verwijderen van het beschadigde voetje van deze 40 pins chip en monteer iets dat beter op elkaar is afgestemd zodat deze kaarten (boards) zonder bruut geweld passen in de betreffende posities.

Een ander probleem in de video sectie wordt gekarakteriseerd door wazige of misvormde karakters. De boosdoeners hier zijn de electrolytische condensators op het Atom printboard. Electrolytische condensators lijken op kleine cilinders (kleine coca cola blikjes of bier blikjes). Zij zijn meestal blauw of grijs van kleur en drie van deze duiveltjes kun je vinden vlak bij chip 6847 de VDG chip. De moeilijkheid met deze condensators is dat zij vaak jong sterven wel soms niet sterven maar langzaam steeds slechter worden. En dat is de moeilijkheid dat langzaam slechter worden dat heb je niet zo gauw in de gaten. Als iets door brandt of kortsluit dan is het veel dramatischer maar je weet het wel meteen. Bij een electrolitische condensator weet je het vaak pas als het zo serieus fout is dat er iets goed mis is.

Een goede oplossing voor zo'n geval is het vervangen van alle electrolitische condensatoren in de machine. En monteer dan geen nieuwe electr. condensators, maar soldeer Tantaal condensators in hun plaats. Tantaal condensatoren zijn veel betrouwbaarder en wanneer zij de geest geven dan gaat dat niet langzaam aan, maar zij scheiden er gelijk mee uit. Ik vind dit een gemakkelijke oplossing. De waardes van alle electr.condensators op het ATOM board is 22 microfarad 16 volt. Deze kunnen vervangen worden door 22mfd tantaal condensators van de zelfde voltage. Denk om de juiste polariteit wanneer u gaat ombouwen. Zowel op print als op de condensator staat aan gegeven wat pos. of neg. is. Foutive montage resulteerd in een dramatische explosie.

Een veel gehoorde klacht is, dat na het aanzetten van de machine het beeld scherm allerlei karakters en symbolen vertoond, totdat de BREAK toets wordt ingedrukt. Ofschoon dit symtoom gewoon is, is het niet normaal. De oorzaak van dit probleem zit in de Reset timing circuit. Wanneer de stroom wordt ingeschakeld of de BREAK toets ingedrukt dan wordt er een reset signaal naar de CPU gestuurd. Dit reset signaal wordt vertraagd door een weerstand-condensator circuit dit is ontworpen met het doel om de voltage gelegenheid te geven te stabiliseren na het inschakelen van d stroom voordat het reset signaal wordt verzonden. Indien de vertragingstijd te kort is dan wordt de reset

verzonden voor dat de zaak is gestabiliseerd en de CPU raakt gefrustreerd. De sleutel voor de oplossing van dit probleem is: het verlengen van de vertraging van het reset signaal. Lokaliseer de condensator en de weerstand deze zitten naast elkaar vlak bij de Atom's luidspreker. De makkelijkste en goedkoopse manier om dit te doen is het vervangen van de weerstand voor een van hogere waarde. De waarde van de originele weerstand is 1K ohm. Vervang deze door een 2K ohm en het probleem moet dan zijn opgelost. Een andere manier is om de condensator te vervangen door een tantaal van een grotere waarde, maar het resultaat is het zelfde uitgezonderd het netto resultaat in je kasboek.

U zult mischien denken dat dit artikel niet lang niet alle mogelijke storingen behandeld die voor kunnen komen. U heeft gelijk. Dat doet het ook niet, maar met de uitzondering van de disk-drive problemen zijn dit toch wel de hoofzaken waar ik mee temaken heb gehad. En ik heb er heel wat gerepareerd.

Onthoud deze sleutel factoren:

-1. Problemen veroorzaakt door slechte contacten zijn de meest voorkomende.

-2. Foute RAM chips zijn de volgende meest voorkomende. Als de defecte RAM in page zero zit dan lijkt het of de machine erg ziek is.

-3. Power supply problemen zijn de eerst volgende. Over en onder voltage veroorzaken veel erratical gedrag.

-4. Slechte condensatoren zijn een verkwente oorzaak van storingen. Zij zouden allen moeten worden vervangen door tantaal condnsatoren.

Zo indien uw ATOM kuren vertoond, check dan deze punten eerst uit. Als het probleem blijft dan heeft u waarschijnlijk wel een probe en of een scope nodig.

Onze Canadese vrienden stellen meer artikelen van technische aard in het vooruitzicht. Als ik ze onder ogen krijg dan zal ik ze vertaald aan de redactie doorgeven.

Wijzigingen in de Teletekst Kaart.

De onderstaande wijzigingen geven u een teletekst kaart, geadresseerd op #BD00 tot #DB0F met bijbehorend Ram gebied van #B400 - #B7FF.

1 Buig alle pennen van de nieuwe 74LS138 uit met uitzondering van 8 en 16. Soldeer deze pennen op pen 8 en 16 van de ACORN VDU kaart IC 5 en lijm het IC daar vast.

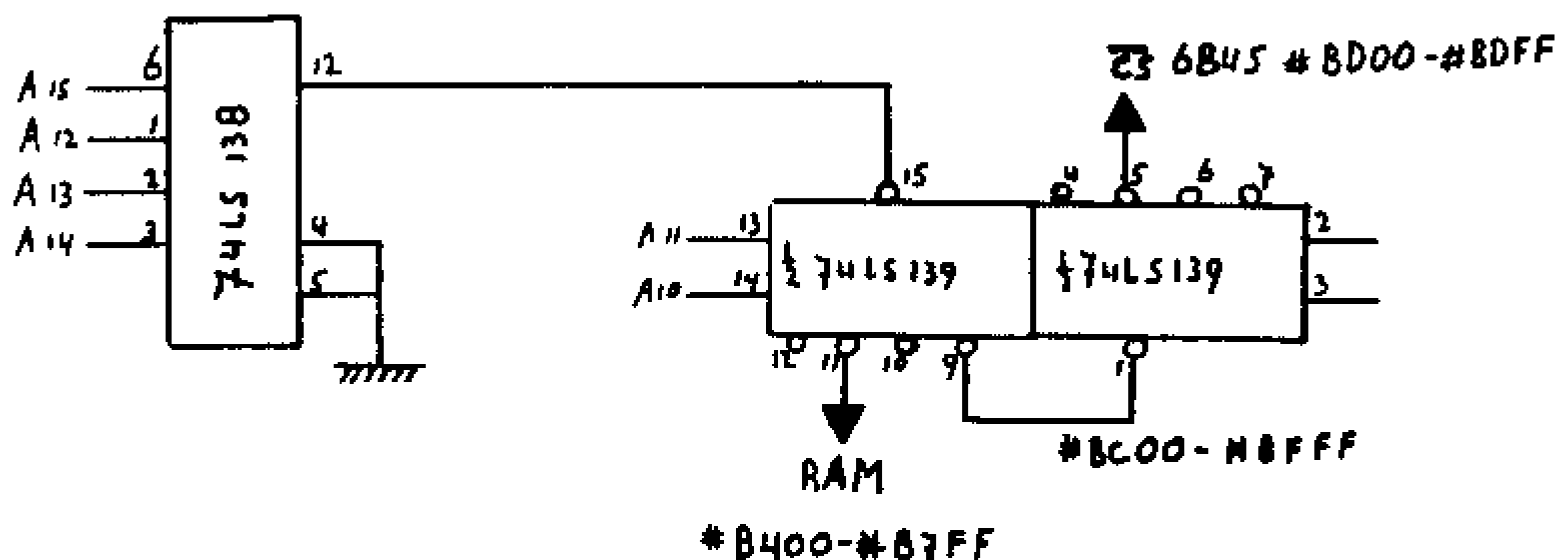
2 Buig de pennen 1, 4, 5 en 15 van Acorn VDU kaart IC 1 uit.

3 Verbind het volgende van de 74LS138:
   a) pen 1 - pen 25 connector A12
   b) pen 2 - pen 24 connector A13
   c) pen 6 - pen 2 connector A15
   d) pen 3 - pen 3 connector A14
   e) pen 4 met pen 5 en pen 8
   f) pen 12 met Acorn VDU IC 1 pen 15.

4 Verbind van Acorn VDU IC 1 het volgende:
   a) pen 1 met pen 9
   b) pen 5 met Acorn VDUC 4 6845 pen 25.

De Teletekstkaart is nu als volgt geadresseerd:

VDU RAM #B400 - #B7FF
6845 CS #BD00 - #BD0F



De keuze van #B4XX als RAM gebied voor de kaart is verre van ideaal. Het gebied #BXXX is immers bestemd als I/O gebied. Waar kan ik het echter anders plaatsen? #0000 tot #9FFF zit vol met RAM, hetzij voor de VDU van de Atom, hetzij gestapeld of als geheugenkaart. Alle andere gebieden dan #B4XX zijn al ergens voor bestemd - vandaar. Big Benny kan

immers eenvoudig elders geplaatst worden, bijvoorbeeld door #B3XX in plaats van #B4XX als CS op PL8 te zetten.
Er is naast deze hardware wijziging tevens een software wijziging nodig om het geheel te laten werken. De software die momenteel in de DOS Eprom zit moet hiervoor gewijzigd worden.

adres	wordt	was	adres	wordt	was
#EF28	#B4	#04	#EE06	#B4	#04
#EEC5	#BD	#08	#EE04	#03	#07
#EECB	#BD	#08	#EE49	#BD	#08
#EED4	#B4	#04	#EE4E	#BD	#08
#EEB0	#B4	#04	#EE52	#BD	#08
#EEB3	#B5	#05	#EE57	#BD	#08
#EEB6	#B6	#06	#EDEA	#BD	#08
#EEB9	#B7	#07	#EDE3	#BD	#08
#EDD7	#BD	#08	#EDDF	#BD	#08

Deze wijzigingen kunnen dan in Eprom gebakken worden. Deze software heeft echter nog een aantal tekortkomingen die door het progamma van F le Blanc (jrg 4, nr. 5, pag. 30) zijn opgelost.
Het bijgesloten programma assembleert deze VDU-soft op de #EXXX pagina nadat eerst deze oude DOS Eprom vanaf #4XXX is overgeladen. Men moet dus Ram hebben op #EXXX. Tevens is er een DOS autoboot programma bijgesloten. Voor de werking hiervan zie jrg.4, nr.2, pag.18.

```
10?#20A=#D5;?#20B= #C2;$256="RUN"
20P.$21;P=#300;[;JSR#FE94;STA#80;RTS;JMP#306;]
30$265="KOEKOEK"
40P.$6$12$15"ACORN ATOM DOS"';K=0
50P.''ENTER PASSWORD: -------"$8$8$8$8$8$8$8
60 T=1;F.I=0TO7;LI.#300;IF?#80<>I?264;T=0
70P."x";N.
80F.I=0TO30;WAIT;N.
90IF T=1;GOTO130
100P.''$7"ILLEGAL PASSWORD."'
110K=K+1;IF K=3;THEN ?#E1=0;P."GET HELP !!!";LINK#306
120P."PLEASE TRY AGAIN."';G.50
130P.''";?#E1=0;P."LOGIN FROM BOOTSTRAP V2."
140F.I=1TO200;WAIT;N.;?#20A=#94;?#20B=#FE
150P.$12"ACORN ATOM DOS."''
160 LINK#C2B2
```

```
10 PROGRAM TELETEXDOS
20
30REM VDU PROGRAMMA F LE BLANC
40REM IN #EXXX GEZET B. KASTEEL
50J=80;@=0
60DIM LLJ
70H=40;REM AANTAL KARAKTERS
80V=25;REM AANTAL REGELS
90F.I=0TOJ;LLI=-1;N.
100F.I=#0000 TO #0FFF
110?(#E000+I)=?(#4000+I)
120N.
130F.I=#ED22TO#EFFF
140?I=0
150N.
160!#E6FB=#FE94FE52
170P.$21
180P=#E00A;[;JMP#ED22;]
190F.I=1TO2;P=#ED22;?#E8=0
200[;:LL1LDX@#0C
210:LL40LDA LL41-#0C,X;STA#200,X;INX;CPX@#1C;BNELL40
220LDA@#7B;STA#200;LDA@#E8;STA#201
230LDA@#E5;STA#206;LDA@#E3;STA#207
240LDA@#20;STA#CD;STA#AC
250LDY@#00;STY#EE;STY#C0;STY#B9;STY#BA
260LDX@#04
270:LL61JSR#E713;DEX;BNELL61
280RTS
290:LL41;]
300!P=#E613E477;P=P+4
310!P=#ECA0EA61;P=P+4
320!P=#EBBCEAF0;P=P+4
330!P=#E89EE953;P=P+4
340[
350:LL0JSRLL52;CMP@#1B;BEQLL56;BIT#B001;BMILL58;BVSLL58
360EOR@#80
370:LL58CMP@#0D;BNELL65;PHA;TYA;PHA;LDY@#0B;JSRLL39
380PLA;TYA;PLA;RTS
390:LL65CMP@#20;BCSLL53;JSR#FFF4
400:LL56JSR#C504;BNELL0
410:LL53RTS
420:LL52PHP;CLD;STX#E4;STY#E5
430:LL4BIT#B002;BVCLL3;LDX@#0A;BNELL51
440:LL3LDX@#01
450:LL51JSR#FB83
460:LL5JSR#FE71;BCSLL5;JSR#FE71;BCSLL5
470:LL54TYA;LDX@#17;JSR#FEC5;LDALL6,X
480:LL6STA#E2;LDALL7,X;STA#E3;TYA;JMP(#00E2);]
490LL7=P+2
500!P=#889AD2DF;P=P+4
510!P=#DFC081E2;P=P+4
520!P=#C6C8D6D8;P=P+4
```

```
530!P=#FDFDFDC2;P=P+4
540!P=#FDEDFDED;P=P+4
550!P=#FDFDFDFD;P=P+4
560!P=#FDFD;P=P+2
570T=LL7-2
580T?3=LL43%256;T?16=LL43/256
590T?5=LL44%256;T?18=LL44/256
600T?2=LL55%256;T?15=LL55/256;[
610:LL44LDY@#0A;STY#BD00;LDA@#80;STA#BD01;LDY@#00;LDA(#DE),Y
620 JMP#FE60
630:LL43AND@#05;ROL#B001;ROLA;JSRLL11
640LDY@#0A;STY#BD00;LDA@#80;STA#BD01;JMPLL4
650:LL55LDA#E7;EOR@#60;STA#E7;BCSLL4
660:LL42JSR#FEFB
670:LL10PHP;PHA;CLD;STY#E5;STX#E4;JSRLL11
680JMP#FE5F
690:LL13PLA;PLA;RTS
700:LL25DEY;BPLLL12;LDY@H-1
710:LL37LDA#CE;CMP@V-1;BCSLL13;INC#CE
720:LL35LDA#CF;SBC@H-1;STA#CF;BCSLL12;DEC#DD
730:LL12RTS
740:LL15DEC#CE;RTS
750:LL24LDY@H;JSRLL14;LDA#DE;STA#CF;LDA#E1;STA#DD;LDA#CE
760BNELL15;LDA@#0D;STY#BD00;LDA#CF;SEC;SBC@#C0;STA#BD01;DEY
770STY#BD00;LDA#DD;SBC@#03;STA#BD01;LDY@H-1;LDA@#20
780:LL17JSRLL16;DEY;BPLLL17
790:LL45LDY#E6;BMILL47;DEY;STY#E6;BNELL47
800:LL46JSR#FE71;BCSLL46;LDY@V-4;STY#E6
810:LL47RTS
820:LL14PHA;CLC;TYA;ADC#CF;STA#DE
830:LL8LDA#DD;ADC@#00;STA#E1;AND@#03;ORA@#B4;STA#DF;PLA;RTS
840:LL16JSRLL14;STY#E1;LDY@#00;STA(#DE),Y;LDY#E1;RTS
850:LL18CLC
860:LL19PHP;ASL#E0;PLP;ROR#E0;RTS
870:LL11CMP@#06;BEQLL18;CMP@#15;BEQLL19
880:LL57LDY#B001;CPY@#3F;BEQLL57;LDY#E0;BMILL20
890CMP@#20;BCCLL21;CMP@#7F;BEQLL22
900:LL33JSRLL16
910:LL34INY;CPY@H;BCCLL23;JSRLL24
920:LL26LDY@#00
930:LL23JSRLL14;STY#E0;LDY@#0F;STY#BD00;LDA#DE;STA#BD01
940DEY;STY#BD00;LDY#E1;STY#BD01
950:LL20RTS
960:LL22JSRLL25;LDA@#20;JSRLL16;BPLLL23
970:LL21CMP@#0D;BEQLL26
980CMP@#0A;BEQLL27
990CMP@#0C;BEQLL28
100CMP@#08;BEQLL29
1010CMP@#1E;BEQLL30
1020CMP@#0B;BEQLL31
1030CMP@#07;BEQLL32
1040CMP@#0E;BEQLL48
```

```
1050CMP@#0F;BEQLL49
1060CMP@#09;BNELL33;BCSLL34
1070:LL29JSRLL25;JMPLL23
1080:LL27JSRLL24;LDY#E0;JMPLL23
1090:LL30LDA@V-1;LDY#CE;STA#CE
1100:LL36CPY@V-1;BCSLL26;INY;JSRLL35;JMPLL36
1110:LL31JSRLL37;JMPLL23
1120:LL28LDA@#20;LDY@#00
1130:LL38STA#B400,Y;STA#B500,Y
1140STA#B600,Y;STA#B700,Y;INY;BNELL38;STY#CF;STY#E0;JSRLL63
1150LDA@V-1;STA#CE;LDA@#B4;STA#DD;LDA#E6;BMILL64;JSRLL48
1160:LL64JMPLL26
1170:LL63LDY@#0D
1180:LL39STY#BD00;LDALL2,Y;STA#BD01;DEY;BPLLL39;RTS
1190:LL32STX#DE;JSR#FD1A;LDX#DE;RTS
1200:LL48LDA@#01;BNELL50
1210:LL49LDA@#FF
1220:LL50STA#E6;RTS
1230:LL2;]
1240B=LL2
1250B?0=#3F
1260B?1=H
1270B?2=#33
1280B?3=#44
1290B?4=#1E
1300B?5=#02
1310B?6=V
1320B?7=#1B
1330B?8=#03
1340B?9=#12
1350B?10=#72
1360B?11=#13
1370B?12=#B4
1380B?13=#00
1390P=P+14;@=0
1400[:LL76JSR#E016;]
1410$P=" OPT ";P=P+L.P
1420[;LDA#2106;JSR#E0FB;JMP#F80B
1430:LL75LDX@#04
1440:LL59LDA LL77,X;STA#100,X;DEX;BPLLL59
1450JSR#E223;LDY@#00;LDX@#00;LDA#2106;JSR#E0FB;BEQLL60
1460:LL62CMP@#02;BCCLL71;BEQLL78
1470JMP#E519
1480:LL78JMP#E50A
1490:LL71JSR#E465
1500:LL60LDX@#FF;TXS
1510:LL72LDA@#2A;JSR#CD0F;JSR#FFF7;JMPLL27
1520:LL77;];!P=544F4F42;P=P+4;?P=#0D;P=P+1
1530[;NOP;NOP
1540:LL74LDA@LL75&#FF;STA#0202;LDA@#EF;STA#0203;BNELL75
1550];@=P
1560P=#E250;[;JSR#E016;]
```

```
1570$P="DR ";P=P+L.P
1580[;NOP;LDA#EE;JSR#F80B;JSR#E016;]
1590$P=" QUAL ";P=P+L.P
1600[LDA#AC;JSR#FFF4;JSRLL76;]
1610N.;P.$6
1620P."EINDE ROUTINE  #"&(Q-1)'
1630IF ?#E8=35;P."OUT OF RANGE"'
1640!#E6FB=#ED6CEE14
1650END
```

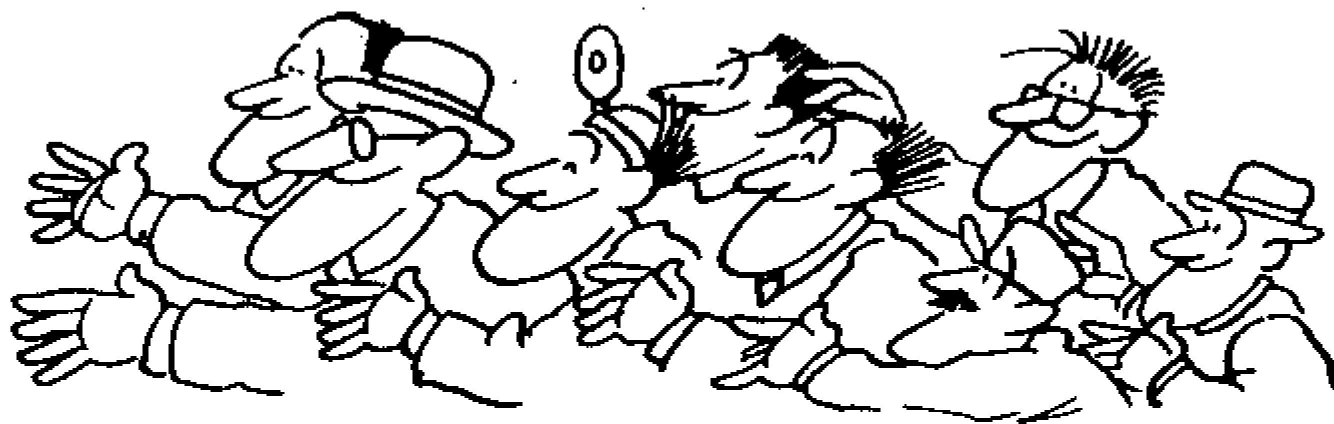
****

ZOMERS (?) NIEMENDALLETJE

```
  1 REM MASTERMIND
  2 REM THEO WAAIJER
  5 P.$2;?#E1=0;@=2
 10 DIMA3,B3,C36,H9,I9,S1
 50 P."**********MASTER MIND**********"'''
 55 P."IK NEEM VOOR U EEN COMBINATIE"'
 60 P."VAN 4 GETALLEN IN MIJN BREIN EN"'
 65 P."U MOET TRACHTEN ZE TE RADEN"''
 70 P.''"                  VEEL SUCCES"
 80 P. '''"U KUNT NU DE MOEILIJKHEIDSGRAAD"'
 82 IN."INVOEREN (0-3) "L;L=L+6
 85 P.$12;?#E1=0;P.'''"DE COMBINATIE BEVAT DE GE-"'
 86 P."TALLEN 0 TOT EN MET "L''
 87 P."U KUNT NU UW COMBINATIE INVOE-"'
 90 P."REN,STEEDS NA ELK GETAL RETURN"'"INDRUKKEN"'
 95 R=0;D=0;E=0;@=0
100 FOR N=0 TO 3
110 J=ABSRND%L
120 A?N=J
130 N.N
140 IFA?0=A?1ORA?0=A?2ORA?0=A?3;G.100
141 IFA?1=A?2ORA?1=A?3;G.100
142 IFA?2=A?3;G.100
200 REM INVOEREN TE RADEN GETALLEN
210 FOR N=0 TO 3
220 INPUT K;GOS.b
230 B?N=K
240 N.N
300 REM VERGELIJKINGSROUTINE
305 P=0;Q=0
310 FOR N=0 TO 3
320 FOR M=0 TO 3
330 IF B?M=A?N;P=P+1
335 NEXT M
340 IF B?N=A?N;Q=Q+1
360 NEXT N;GOS.c
370 P.$12"U HEEFT "P" CIJFERS GOED"'
380 P."WAARVAN "Q" OP DE JUISTE PLAATS"'
385 R=R+1;IFR>9;G.550
```

```
390 IF Q<4;;GOS.d;G.210
400 P."U HEBT HET GERADEN IN "R" KEER"'
410 IN."NOG EEN KEER (J/N)"$S
420 IF $S="J";G.80
430 P.$12''''''"OKE DAN STOPPEN WE ERMEE"''"TOT ZIENS"
440 END
500bD=D+1;C?D=K;R.
510cE=E+1;H?E=P;I?E=Q;R.
520dF.F=4TO D STEP4
522 F.G=-3TO0;P.C?(G+F)," ";N.G
530 P." - ",H?(F/4)," "I?(F/4)'
540 N.F;R.
550 P.$12;?#E1=0
560 P.''''"JA HOU MAAR OP"'
570 P.''"DIT IS HUILEN MET DE LAMP AAN"'
575 P.''"DRUK OP SPATIEBALK "';LINK#FFE3
580 P.'''"HET WAS "A?0,A?1,A?2,A?3';GOS.d
590 G.410
```

```
100REM INTERRUPT-MUZIEK
110
120PASS0;GOS.a
130PASS1;GOS.a
140END
150
160a ASM-BEGIN
170
180 .TABLE #7000,#8000
190 .CODE #3000
200 .LIST .OPTION #40
210\VARIABELEN
220:LSP = #B002
230:MASKER=4
240:INTVEC=#204
250:TIMERMODE=#B80B
260:INT'MASK=#B80E
270:TIMERL=#B804
280:TIMERH=#B805
290:FVARL=#327
300:FVARH=#342
310
320:init
330 LDA@(toon)%256 ;STA INTVEC
340 LDA@(toon)/256 ;STA INTVEC+1
350 LDA@#00 ;STA TIMERMODE
360 LDA@#C0 ;STA INT'MASK
370 LDA@10 ;STA FVARL ;STA FVARH
380 STA TIMERL ;STA TIMERH
390 CLI ;RTS
400
410\GELUIDROUTINE
420:toon
430 LDA FVARL ;STA TIMERL
440 LDA FVARH ;STA TIMERH
450 LDA LSP ;EOR@ MASKER ;STA LSP
460 PLA ;RTI
470
480 .SYMBOL
490 .END ASSEMBLER
500 R.
```



```
*********************************************
*  SOFTWAREPAKKET voor onderwijs ELECTRONICA *
*********************************************
```

door Jan Bronzwaer

Als leraar verbonden aan een middelbare technische school en als fanatiek ATOM-gebruiker opgegroeid , kon ik het niet laten mijn computer te gaan gebruiken voor mijn werk .
In de 1e klas verrichten "mijn" leerlingen metingen aan componenten uit het electronica vakgebied en nemen van die componenten de karakteristieke eigenschappen op in grafieken . Hiervoor heb ik opdrachten geschreven (waarvan ik aan geinteresseerde clubleden gaarne kopieen wil verstrekken).
Via de ATOM en dit pakket wordt hen nu de gelegenheid geboden om de uit de metingen verkregen meetresultaten op nauwkeurigheid te controleren , aanvullende informatie te verkrijgen en hardcopy's te maken van de grafieken .
Een en ander is , hoewel nog lang niet volledig , al een jaar in gebruik , tot tevredenheid van alle partijen . De programma's zijn "leerling-vast" , dus zo gebruikersvriendelijk mogelijk geschreven . De praktijk heeft bewezen dat men , zonder hulp van mijn kant , met de software kon werken.

Vanuit de filosofie "houd niet alles voor uzelf , een ander lust het toevallig" bied ik U dit pakket(je) hierbij aan .
Leden van de regio Limburg kunnen het vinden in ons Diskarchief ACL2 op schijf nummer 3 . Het pakket bestaat uit 9 files , te weten:

EM1	: de bootfile , waarmee U opstart.
MENU	: het diskette-menu
7OUTIL2	: een 4k utility voor P-charm
EM1A	: programma over de wet van OHM
EM.2B	: programma over temperatuurscoefficienten
EM.3A	: programma over de belastingslijn
POTM.1	: programma over een potentiometer
POTM.2	: idem
EM.6b	: programma over spanningsbronnen

De programma's zijn allen geschreven in P-CHARME in een zoveel mogelijk gelijke structuur . Mensen die niet zozeer geinteresseerd zijn in het inhoudelijke maar zich wel willen begeven op het P-CHARME-gebeuren vinden in de listings en het gebruik van extra statements met 7OUTIL2 mooie voorbeelden .
Wat heeft U nodig om dit pakket te kunnen draaien ?

1. Een drive.
2. Een Atom met P-CHARME op het A-blok.
3. Geheugen van #2600 - #A000 (normaal uitgebreid , dus).
4. Eventueel een STAR-printer , (kom ik op terug!).

EM1

Het softwarepakket wordt opgestart met *EM1.
De bootfile kijkt of P-CHARM voorstaat en waarschuwt U als dit niet het geval is. In het andere geval wordt MENU opgestart .

MENU

Het diskettemenu dirigeert U naar het gewenste programma en bevat bovendien een 13 -tal pagina's handleiding . Gestart wordt met een plaatje dat U natuurlijk naar wens kunt aanpassen. In de listing ziet U waar dit plaatje moet staan.
Bij keuze van een bepaald programma verlaat MENU zichzelf en het gekozen programma wordt vanzelf gehaald en opgestart. Vanuit elk programma kan men terugkeren in MENU.

7OUTIL2

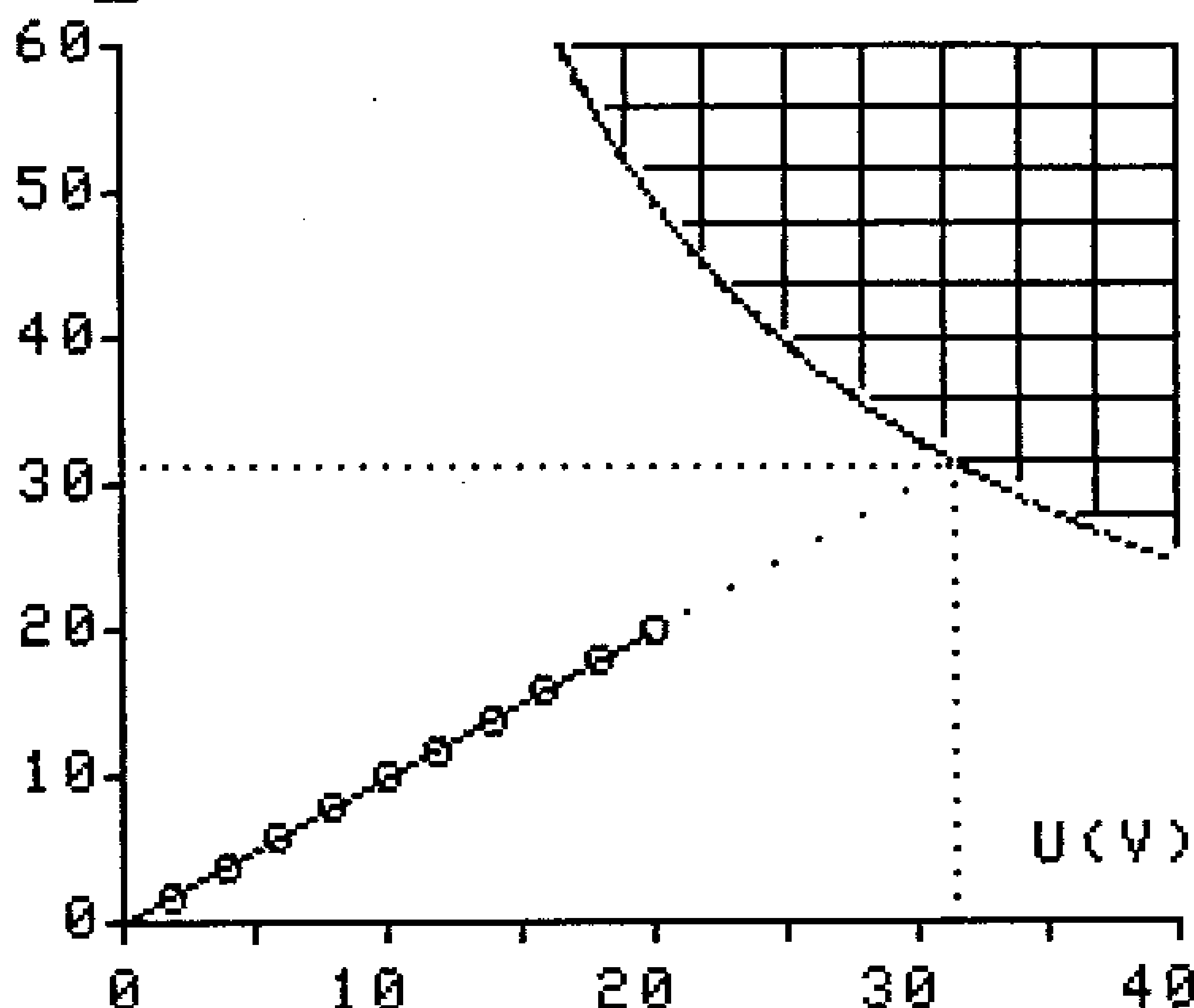
Dit is een 4 kilobyte lange Utility van #7000 - #8000 met extra statements , die in de programma's worden gebruikt . Een lijst van deze statements krijgt U op het scherm door intypen van het commando "VLIST" . 7OUTIL2 wordt door MENU vanzelf ingeladen en #3FC wordt op #70 gezet . De diverse programma's draaien dan ook alleen als 7OUTIL2 zich in het geheugen bevindt. Op de reeds eerder genoemde diskette van het diskarchief staat ook de source van deze utility onder de naam "70-SRC2". Het grootste deel van de statements is beschreven in mijn reeds eerder gepubliceerde artikel "22 extra statements" in Atom Nieuws 5-3 pagina 16 met dit verschil :
BLOCK-CUBE-NOSNOW-SNOW-CURSOR-HEADER-SPEED : verwijderd
CIRCLE : laatste snelle versie
ENTER : verbeterd , zie BGE nr.2-1987 NINPUT
PSCREEN-GSCREEN : verbeterd , zie BGE 1-1986
Toegevoegd : GDUMP , 'n screendump voor STAR-printers (Joop Engels). Met dit statement maken we in elk programma een screendump van de getekende grafieken . Beschikt U over een andere printer , dan moet U GDUMP in de source door een aan uw printer aangepaste routine vervangen.

DE PROGRAMMA'S
==============
Van de programma's , die allen dezelfde structuur laten zien , beperk ik mij tot een beknopte omschrijving van het doel en de grenzen , waaraan de invoer van gegevens is gebonden . Dit laatste was o.a. noodzakelijk omdat de grafieken vaste assenverdelingen hebben en het niet zinvol is om met getallen te rekenen die buiten deze assenverdelingen vallen . U kunt natuurlijk zelf de grafieken intelligent maken , maar denk daarbij aan het begrensd geheugen .
Bij elk programma behoort in feite een meetopdracht . Voor mensen die hierin zijn geinteresseerd staan kopieen ter beschikking . Gelieve in dat geval telefonisch contact met mij op te nemen . (04459 - 1969)

EM1

Dit programma behandelt de Weerstandslijn en dissipatiehyperbool van een lineaire weerstand en tekent dus I=f(U). Invoer alsvolgt begrensd:
WEERSTAND : alleen E-12 reeks van 330 - 1000 ohm
VERMOGEN : 0.3 - 1 watt
KLASSE V-meter : 0.25 - 5%
KLASSE A-meter : 0.25 - 5%

Voorbeeld grafisch resultaat :

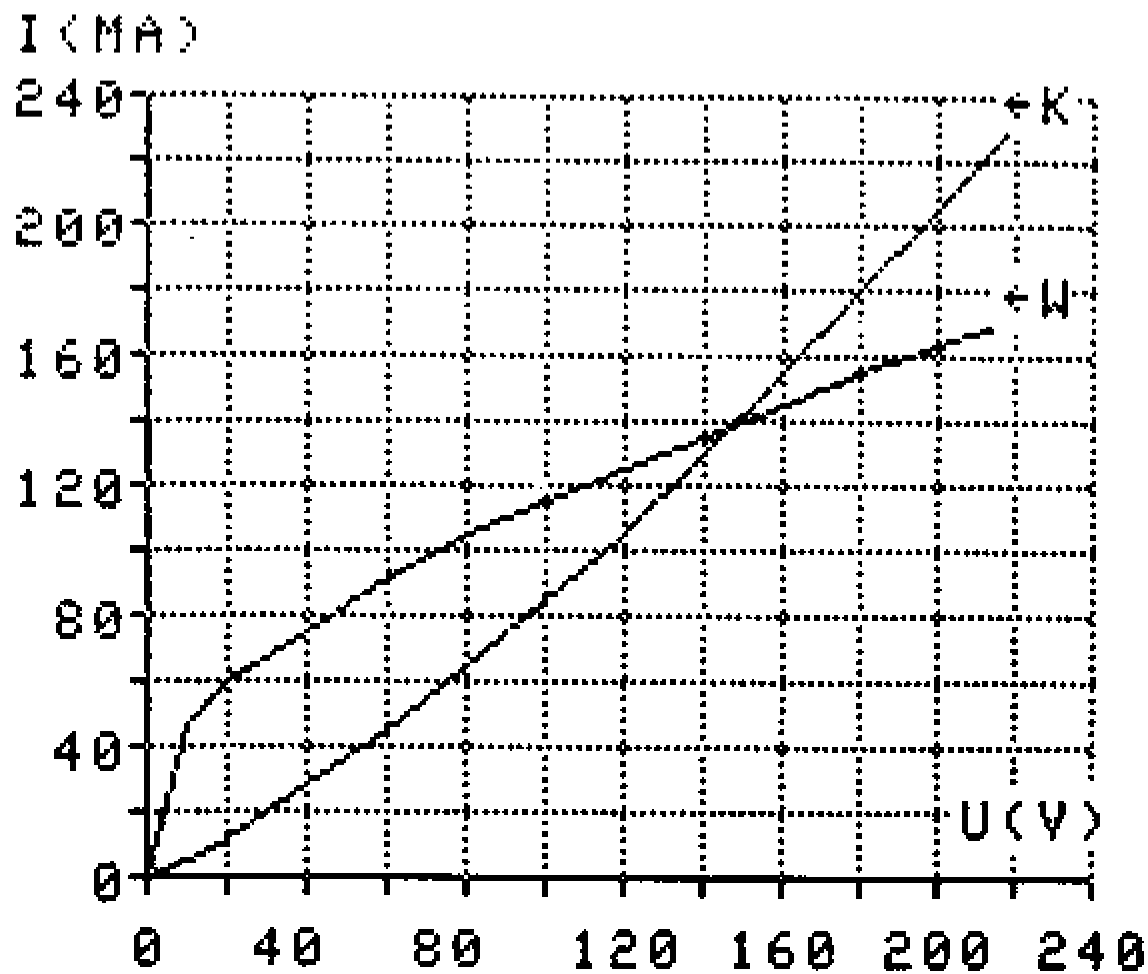


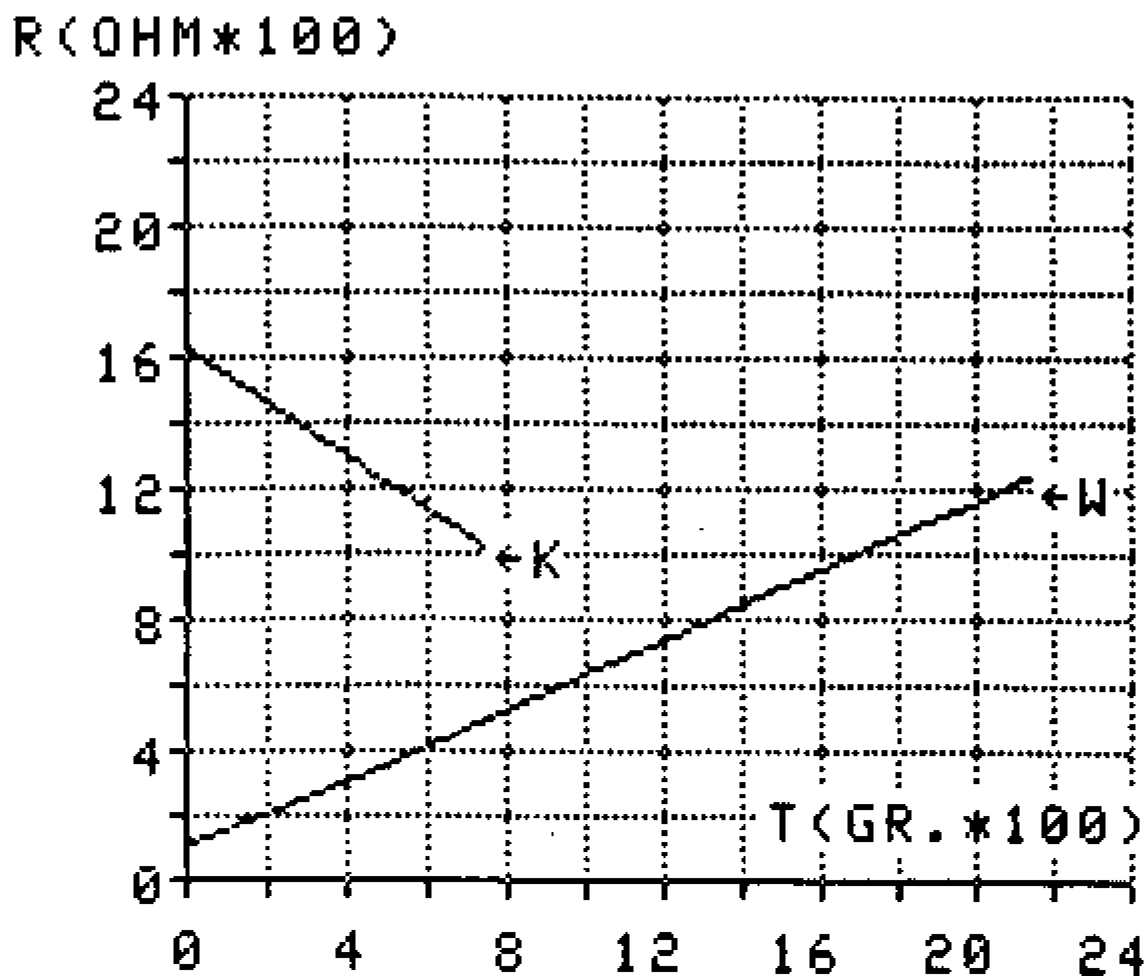
EM.2B:

Dit programma behandelt de temperatuurscoefficient van wolframdraad (positief) en kooldraad (negatief). Dit is met gloeilampen het gemakkelijkst te onderzoeken. Het tekent de I=f(U) en R=f(T) karakteristieken m.b.v. benaderingsformules.Het programma produceert uitvoerige tabellen . Hierin dienen de meetresultaten te worden getypt en de berekende temperaturen volgen meteen. Verder moeten de met een ohmmeter bepaalde koudweerstanden worden ingevoerd.

Begrenzing invoer :
KAMERTEMPERATUUR : 10 - 29 graden Celcius
R(koud) WOLFRAM : geen grens (normaal ca. 180 ohm)
R(koud) KOOLdr. : geen grens (normaal ca. 1800 ohm)

Voorbeeld grafisch resultaat :

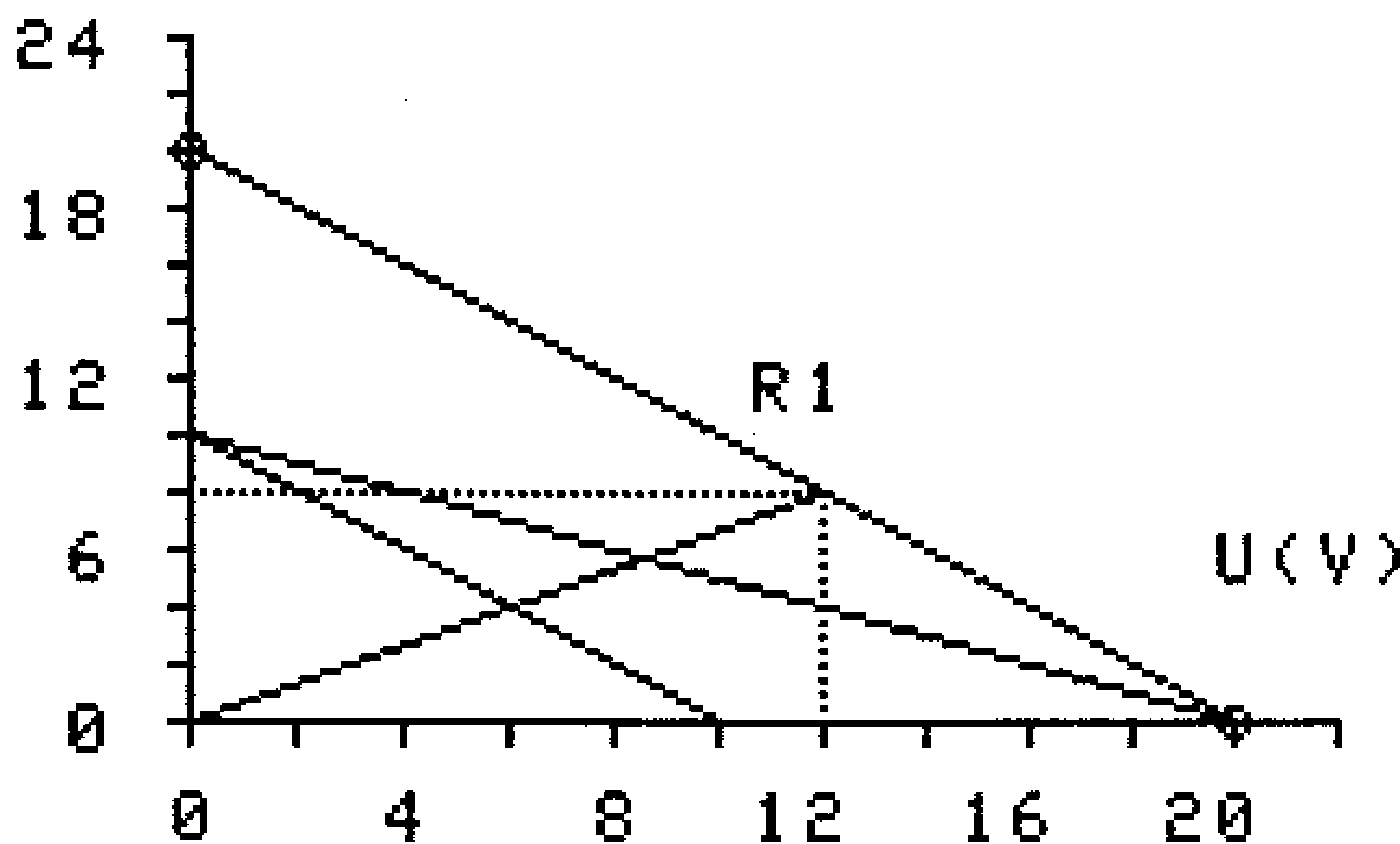




EM.3A

Dit programma gaat in op de belastinslijn , die ontstaat als men een spanningsbron met een bepaalde inwendige weerstand gaat belasten met een variabele weerstand . Dit onderwerp is van belang bij het bepalen van instelpunten van schakelingen die niet-lineaire weerstanden of halfgeleiders bevatten .
Ook hier moeten meetresultaten worden ingetypt ter vergelijking van metingen met zuivere berekeningen .Begrenzing invoer :
VOEDINGSSSPANNING : 0 - 22 volt
WEERSTAND (R1) : minimaal VOEDINGSSPANNING / 24 mA

Voorbeeld grafisch resultaat :



U(B) = 20.00 VOLT
R1 = 1000.00 OHM
NIET BELAST : U=U(B);U(R1)=0
KORTGESLOTEN : I(K)=U(B)/R(1)

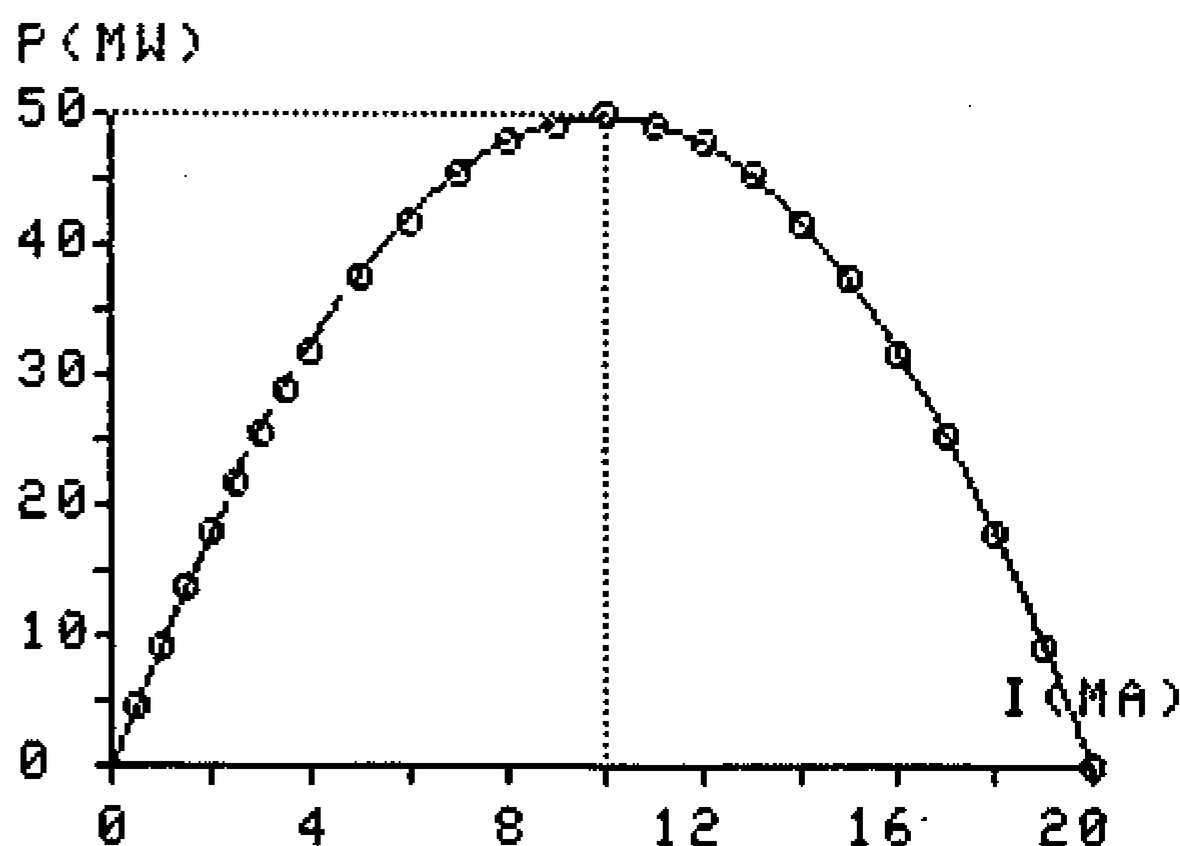
POTM.1 en POTM.2

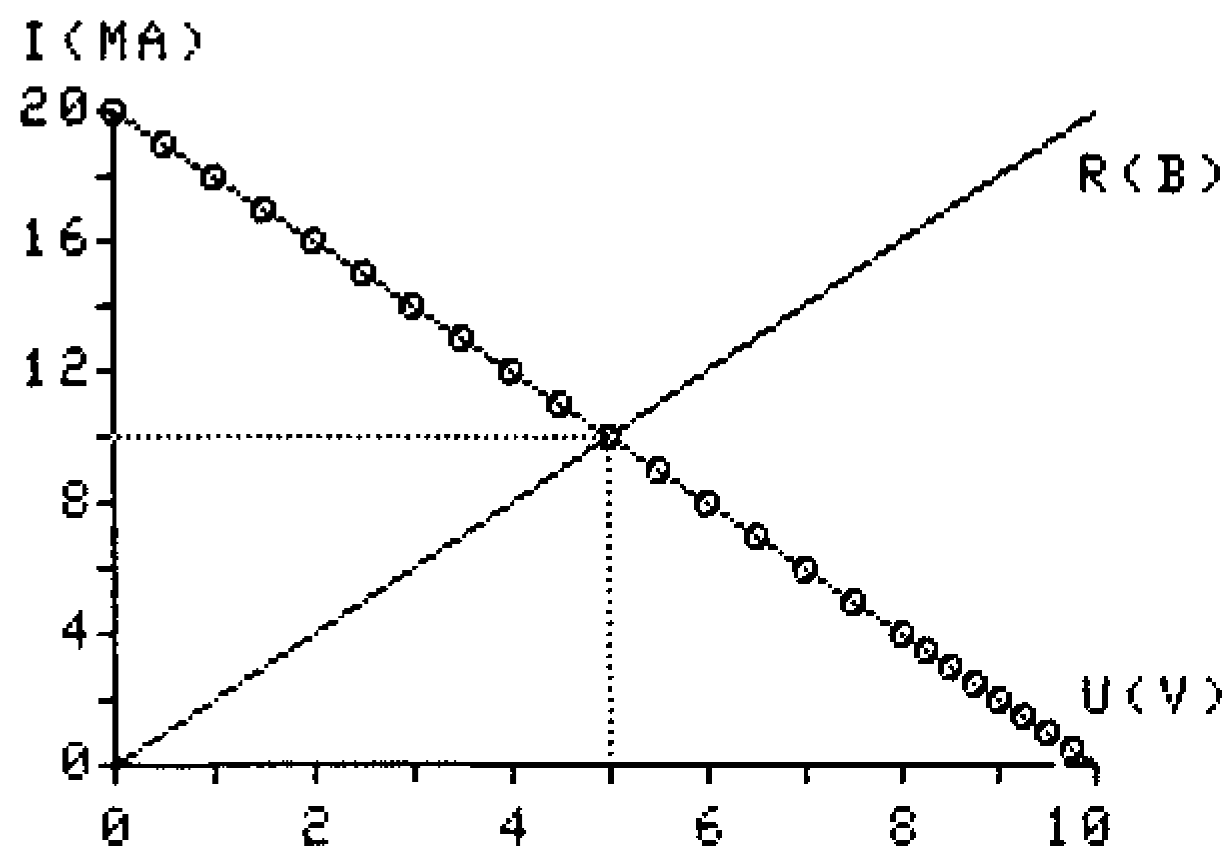
Deze twee bij elkaar behorende programma's zijn herschreven versies van de reeds eerder gepubliceerde programma's :
POTMETER in Atom Nieuws 4-7 en POTENTIOMETER deel 2 in Atom Nieuws 5 - 3 . U zij naar deze artikels verwezen .

EM.6B

Dit programma behandelt de karakteristieken , zoals die behoren bij een spanningsbron met een bepaalde inwendige weerstand . Van de ingevoerde bron wordt de vermogenskarakteristiek P=f(I) en de belastingskarakteristiek I=f(U) getekend zoals in onderstaande voorbeelden is weergegeven . Begrenzing invoer :
E.M.K. VAN DE BRON : 0 - 10 volt
INW.WEERSTAND (R1) : minimaal E/0.02 en maximaal E/0.0005

Voorbeeld grafisch resultaat





Atomaire wetenswaardigheden

In de loop der tijd dringen de verschillende, soms unieke mogelijkheden, van de ATOM tot je door. Het leek mij geen onaardig idee daar eens iets over te schrijven. Ik ga uit van een standaard ATOM-configuratie, met floating pointvariabelen op #2800 en een 'normale' diskdrive met ATOM DOS en daarbij DECADOS. Verder werk ik onder BRANQUART, maar dat laatste is niet van wezenlijk belang.

TEXT - *EXEC
Na het commando TEXT uit de wordpackrom wordt alles wat op het scherm verschijnt als een textfile vanaf #2800 opgeslagen. Dit zelfde kun je doen met een listing van een programma. Houd daarbij wel rekening met het feit dat de textfile meer ruimte in beslag neemt per regel dan de programmafile. Plaats het programma daarom bijvoorbeeld eerst in het hoge geheugen voordat je het na TEXT en met LIST wegschrijft naar #2800. Na EDIT en Old Text? Y, kan deze listing worden bekeken. Je kunt daarbij gebruik maken van de zoekfunkties en de mogelijkheid om terug te bladeren. Het is ook mogelijk om veranderingen aan te brengen met behulp van de texteditor. Echter is in de wordpackrom geen commando beschikbaar waarmee van een textfile weer een gewone listing gemaakt kan worden. Dit commando zit wel in de DISK-rom. Schrijf eerst de textfile van de listing naar diskette. In de textfile mogen alleen regels uit de listing staan. Met *EXEC "textfile" wordt de textfile weer teruggemuteerd naar een normale listing. Op deze manier kun je de wijzigingen in de programmafile op een eenvoudige en overzichtelijke manier manier aanbrengen. Invoer van regels langer dan 64 karakters is op deze manier ook mogelijk geworden.

De Disk catalogus.
Het gebeurt wel eens dat door een of andere storing of een foutieve opdracht de catalogus van een diskette wordt vervormd. Veel zaken, zoals verkeerde qualifiers kunnen worden recht getrokken door gebruik te maken van de commando's uit de DISK-rom. Een probleem als het foutief weergeven van het aantal beschikbare sectoren is met behulp van de DISK-rom echter niet te herstellen. In het geval de schijf geformateerd is met 400 sectoren, moet op adres #2106 de waarde #01, en op #2107 de waarde #90 staan. Is dit niet het geval dan moeten deze waarden daar eerst gepoked worden voordat de herstelde catalogus weer kan worden weggeschreven. Dit gebeurt indien JSR#E75B en JMP#E74A na elkaar worden uitgevoerd. Op deze manier kunnen ook andere veranderingen in de catalogus worden aangebracht.

Disk - Cassette vice versa.
Voor het saven van programma's vanaf diskette naar tape bestaan verschillende programma's. Ik heb hiervan het mooiste programma uitgekozen, namelijk dat van Ronald Boers, en dit op #8200 geplaatst. Het werkt onder P-charme en Disk-rom. Indien je eveneens de Josbox (AXR-1) geinstalleerd hebt, dien je even op te passen dat niet de FCOS uit de Josbox wordt aangeroepen. Dit is te voorkomen door voor FCOS, het statement DISK op te nemen. Als laadgebied heeft BACKUP nu het gebied vanaf #2900 en verder tot

zijn beschikking. Eventueel zou je tot #2400 omlaag kunnen. Lager gaan is niet aan te raden omdat bij het inladen van een file van diskette het buffergebied vanaf #2200 onderhevig is aan veranderingen. Het gebied vanaf #2900 lijkt mij echter wel voldoende.
Heb je geen schakelsoft dan kun je het programma van Ronald Boers nog wijzigen voor P-charme. Daarbij is het onderstaande tabelletje met vectoradressen van belang.

	Josbox/AXR-1	P-charme	Disk
SAVVEC	#A5E7	#FAE5	#A576
BGTVEC	#A527	#ADE0	#A590
BPTVEC	#A592	#AEDE	#A5F1

Verander bij een programma dat lager staat dan pagina #29 nooit de eerste regel met REM in PROGRAM, want dan wordt het gebied van #28BD tot #28FF op nul gezet ten behoeve van eventuele P-charme procedures en funkties, en wordt een deel van je programma vernietigd. Met deze mogelijkheid moet altijd rekening worden gehouden als er met PROGRAM gewerkt wordt.
Let er bij het saven van adressen naar diskette altijd op dat je het eindadres aangeeft met TOP + 1, of laatste byte programma + 1. Bij het backuppen naar tape zou anders het laatste byte verloren gaan. Dus bijvoorbeeld #3C00, en niet #3BFF. Dat het bij diskette wel goed gaat, komt doordat altijd gehele pagina's worden weggeschreven, dus tot en met het adres ..FF, en geen gedeelten daarvan, al is misschien wel een gedeelte opgegeven.

Nu dan het backup gebeuren van tape naar diskette. Na wat wijzigingen en/of toevoegingen is het mij gelukt in totaal drie casdis programma's aan het werken te krijgen. Dat zijn de programma's CASDIS, CAS>DIS en CASDIS3. Het laatste is een gewijzigde versie van het door Charl de Moor geschreven programma. (A.N.5-2) Dit programma werkt nu onder de DISK-rom. Van de overige programma's zijn mij de auteurs niet bekend. Het programma van J.Klein (Atomix nr.22 of 23) heb ik niet kunnen testen omdat ik het niet bezit. Allicht werkt een van deze programma's ook op jouw ATOM. Mijn voorkeur gaat uit naar het programma CAS>DIS, dat in basic geschreven is voor de P-charme. Het werkgebied begint, evenals bij de andere programma's, op pagina #29. Het programma staat in het hoge geheugen en maakt gebruik van procedures en een funkie. Het is vanuit elke textpage te starten door CAS>DIS gevolgt door <return> in te tikken. Het grote voordeel van dit programma is dat alleen volledige files van de band worden gehaald. Dit is van belang als er zich korte pauzes tussen de op de tape gesavede files bevinden. Een klein nadeel is dat, indien van de tape files worden gehaald waarvan de eerste zeven karakters van de naam overeenkomen, de als eerste gesavede file wordt overschreven.

P-charme uitbreidingen.
Zoals bekend is het mogelijk om op tamelijk eenvoudige wijze de P-charme uit te breiden met extra statements en funkties.
De tabel met de statements moet altijd aan het begin van een pagina staan, en beginnen met FF E3 C6, en worden afgesloten met 80. (hexadecimale waarden) Het hoge byte van de executieadressen van de aan te roepen statements wordt opgehoogd met #80, voordat

het in de tabel wordt geplaatst. Om deze reden kan een stuk assembler in het geheugen vanaf #8000 niet direkt worden aangeroepen. #80+#80=#FF ...! Indirekt zou wel kunnen via een jump in het lage geheugen. Om de statements aan de P-charme bekend te maken moet hat adres #3FC gevuld worden met de pagina waar de tabel staat. Meerdere tabellen met statements kunnen gelijktijdig aan de P-charme bekend worden gemaakt. Daartoe wordt het eerste byte van de tabel op het paginanummer waarmee #3FC is gevuld, gevuld met het paginanummer van de volgende tabel. De laatste op te nemen tabel begint weer met #FF.

Op dezelfde manier als er een uitbreidingstabel met statements gemaakt kan worden, is het mogelijk een tabel met funktieuitbreidingen te maken. Deze tabel begint met FF E3 C7 en eindigd eveneens met 80. Verder moet ook P-FUN geplaatst worden aan het begin van een pagina. In #3FA wordt het paginanummer van de tabel opgeslagen. In #3FB moet #B3 komen te staan, in #3EE moet 00 staan. In #3EF komt het paginanummer waarop de assembler van PFUN begint. Indien een ERROR 29 zou ontstaan, wordt nu eerst gesprongen naar PFUN, een interpreter, die er voor zorgt dat de tabel met funkties wordt uitgelezen. Voorwaarde is wel dat de funktieuitbreidingen worden aangeroepen in een programma dat begint met ' PROGRAM'! Problemen kunnen echter ontstaan bij het gelijktijdig gebruiken van deze funktieuitbreidingen en de FUNKTION uitbreiding in een programma. Bij een ERROR 29 wordt namelijk alleen gekeken naar de funktieuitbreidingen in de tabel en niet die in het programma. Dit is op te lossen door het adres 3FB te vullen met bijvoorbeeld #FF, zodat de tabel niet meer herkend wordt, en er weer gekeken wordt naar de FUNCTION uitbreidingen.
Met name een stringhandler is een nuttige funktieuitbreiding. Deze heb ik met PFUN opgenomen achter BRANQUART. De initialisatie van BRANQUART verzorgt het zetten van de vektoren voor de funktieuitbreidingen en zoekt tevens naar een tabel die begint met E3 C6, zodat de statement uitbreidingen dierkt kunnen worden aangeroepen, zonder dat eerst het adres 3FC met de hand moet worden gevuld.

Nieuwe TOOLBOXEN.
Het CLS commando uit GAGS-2.3 werkt goed voor de 80 koloms kaart. Het CLS statement uit GAGS-2.0 werkt daar echter niet. De statements SNOW en NOSNOW werken niet in combinatie met GAGS-2.3. De PAINT routine uit GAGS-2.3 heeft andere startwaarden nodig dan die uit GAGS-2.0 om hetzelfde resultaat te bereiken.
Door mij zijn een tweetal nieuwe toolboxen samengesteld, namelijk ACE-3.0, Atom Commands Extension met commando's, en ASE-2.0, Atom Statements Extension met statements en een aantal commando's zoals SEARCH die ook in een programma kunnen worden gebruikt. ACE-3.0 heeft een vrij definitieve versie aangenomen en draait al sinds juni 1985. ASE-2.0 is nog niet zo lang geleden in zijn huidige versie gereed gekomen. In ASE-2.0 is een door Wim Osterholt verbeterde versie opgenomen van GRMOD, die nu niet meer op P.$0 stukloopt. **Tevens zijn een aantal door Rijn Buve nieuw ontwikkelde statements opgenomen. Het geheel is gecomplementeerd met statements uit andere boxen, die in mijn systeem hierdoor overbodig werden.
De opzet van beide boxen is om de commando's en de statements van elkaar te scheiden, en het aantal in een schakelsysteem op te**

nemen boxen zo sterk mogelijk te beperken. Met behulp van de P-charme, Gags-2.3, de DISKrom en de ASE-2.0 zijn vrijwel alle programma's te draaien. De Wordpack, Calcrom, ACE-3.0 en Salfa-2.5 kunnen daarnaast als toolboxen worden gebruikt.
De aan te roepen statements en commando's uit verscheidene boxen kunnen via het statement COM worden uitgelezen. Werking : COM <romnummer>. Bij de P-charme worden tevens de uitbreidingen op het scherm afgedrukt. Ikzelf heb dit statement achter BRANQUART geplaatst en in de tabel van de bij BRANQUART horende statements opgenomen, door de volgende regel in de source van BRANQUART toe te voegen: 4731.ASCII /COM/,#FE,#15,#77. Het object van COM staat bij mij dus vanaf #1577. Het is natuurlijk ook mogelijk COM in een P-charme uitbreiding op te nemen. Dit is bijvoorbeeld gedaan in de eveneens door mij samengestelde P-charme uitbreiding PE-BFFF. De uitvoer na COM 7 ziet er als volgt uit:

```
ROM 7: eprom
LEFT    RIGHT   MID     STRING
"       $       TAB     SPC
TIMER   BINARY

MINIAS  EXVAR   FON     FOFF
COM     ATTACH  EXTERN  p
CTRL    LRG     WINK    SYMBOL
PAG     HEPR    PLIST   DUMP
FLASH   GINVERT SWAP    REFLECT
SIZE    PE

RENUM   AUTO    CONT    DEL
READ    PEND    <       CEND
WHILE   WEND    XIF     ELSE
FEND    BEEP    ON ERROR ON
CASE    COPY    ICOPY   POP
KEY     INKEY   ZERO    VTAB
HTAB    DATA    FUNCTION PROC
PROGRAM PAUSE   BSAVE   VAR
HEX     CAT     COS     STOP
RESTORE GOTO    GOSUB
```

Het schaduwbyte is #9FFF, terwijl het schakelbyte #BFFF is, vandaar de naamgeving PE-BFFF. Van de vermelde funkties en statements zijn de meesten wel beschreven in vorige uitgaven van ATOM NIEUWS.
Enige toelichting:
MINIAS stelt de mogelijkheden van de miniassembler tot je beschikking. (A.N.3.4-68, A.N.3.7-88, A.N.4.7-32) Na EXVAR kunnen in een P-charme programma symbolische variabelen gebruikt worden. De variabelen worden gedefinieerd door ze een waarde toe te kennen, bijv: "TESTVAR=PI*10". De variabelen worden opgeslagen in het hoge gestapelde geheugen vanaf #9800. Na gebruik van het 'PROGRAM' statement worden alle oude variabelen vergeten. Helaas werken de variabelen niet in FOR-NEXT, Procedures, funkties en input's. Met FON en FOFF komen de funktieuitbreding al of niet tot je beschikking. Hiertoe behoort onder andere de stringhandler. Na LRG kunnen 255 karakters na elkaar worden ingevoerd. Met SYMBOL en <CTRL A> kunnen grafische karakters met behulp van het toetsenbord worden ingevoerd. PAG <var> bepaalt het aantal regels dat naar de printer wordt verstuurd. Na een aantal linefeeds en een toetsaanslag wordt het printen voortgezet

tot het opgegeven aantal weer is bereikt. Van de overige statements en funkties zal het gebruik wel duidelijk zijn, of is wel een beschrijving te vinden.

Twee programma's
Het programma SHAKER is een voor P-charme en ASE-2.0 gewijzigde versie van het programma COCKTL van Rijn Buve. Het programma COCKTL draait alleen onder de AXR2. Dit programma is je behulpzaam bij het maken van cocktails. Een extra funktie zit achter de niet beschreven letter 'p'. Probeer maar eens.
Het tweede programma vergt nogal veel van je computersysteem. Het gaat er vanuit dat je P-charme met stringhandler hebt, in combinatie met ASE-2.0 en ACE-3.0. Foutieve invoer is niet mogelijk. Het resultaat is uiteindelijk de tijd in de rechterbovenhoek van het clear 0 scherm.

Gerard van den Brandhof,

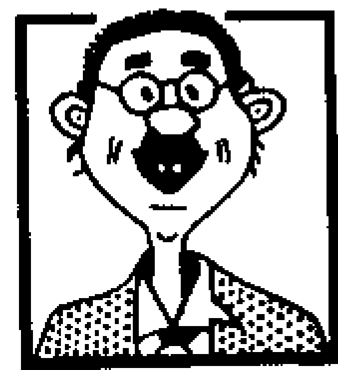
BUS PROBLEMEN

Als je de uitbreidingen voor de ATOM in een 19" rack hebt zitten, dan kan dat problemen geven met de I/O's van de 6522. Dit is afhankelijk van de lengte van de kabel plus backplane. Bij mij is dat ongeveer 65 cm. De problemen bij mij uiten zich in het niet werken van de eprom programmer. Verder werkt alles wel naar behoren, d.w.z. de schakelkaart, de 16K geheugenkaart en de disk-controller kaart. De oorzaak is overspraak tussen de draden in de flatcable, vooral als de 6522 in de input mode staat. De ingangsweerstand is dan n.l. 7K. Het een en ander is met een oscillograaf te zien als grote stoorpulsen op het gewenste signaal. De oplossing die ik gekozen heb is als volgt. Ik heb de aansluitplug gescheiden in twee pluggen. De a-rij 3 t/m 22 en 1 en 32 op een aparte connector boven op de ATOM. De b-rij plus 25,26,28,29 en 30 van de a-rij blijven naar het 19" rack gaan. Van deze flatcable worden dan de niet gebruikte aders aan de massa gelegd. Het soort connector is niet zo belangrijk, ik heb een a-c connector gebruikt. Zie figuur 4.
Zie ook het artikel busprints in A.N.4-4 blz.8.
Bovendien is het nu ook mogelijk tegelijk een busuitbreiding, b.v. de originele ATOM diskdrive, de eprom programmer te gebruiken.
De eprom programmer werkt nu dan ook weer perfect, zelfs ook voor de 2716 en 2764. Maar dat is voor de volgende keer.

Wordpack tip.

Als de wordpack wordt ingeschakeld met het EDIT commando en je houdt de returntoets te lang ingedrukt, dan is het alsof na de vraag "Old Text?" al de return is ingedrukt zodat de vraag "Old Text" helemaal niet verschijnt. Dus na het EDIT commando de returntoets heel kort indrukken.

Sjaak Geene



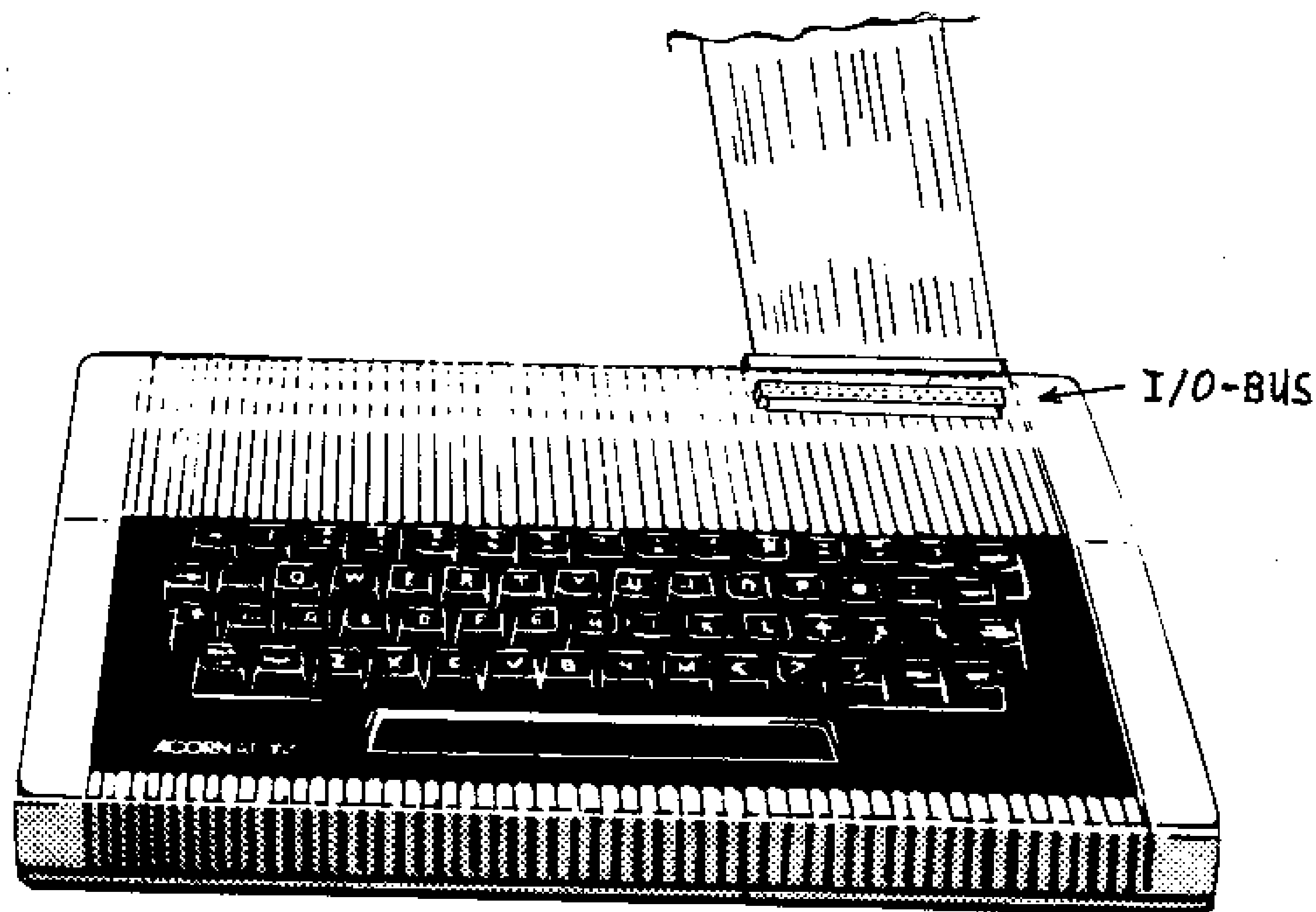
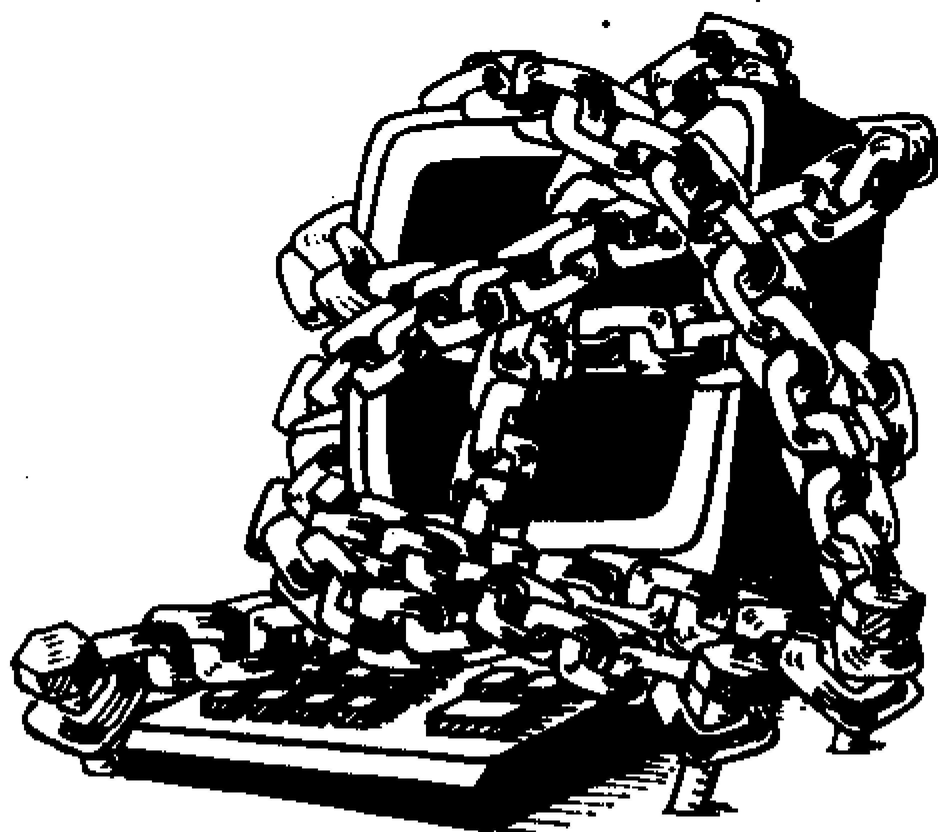


fig 1.

Nu volgt een kleine onderbreking voor de lange winteravonden.

```
 10 REM SNEEUW
 20 C=2;H=96;B=128;REM C=MODE,H=HOOGTE,B=BREEDTE
 30 DIM L255,Q255,S255,R100
 40 P.$21;DIMLL10;FORI=0TO10;LLI=-1;FORI=0TO1;[
 50 :LL0 LDX@H
 60 :LL1DEX;LDAL,X;STA#5A;TAY;STX#80
 70 DECQ,X;LDAQ,X;CMPS,Y;BCCLL5;BNELL2
 80 STX#80;LDX#5A;INCS,X;JMPLL5
 90 :LL2STA#5C;INC#5C;STX#80;JSRLL10;DEC#5C;JSRLL10
100 :LL5LDX#80;BNELL1;RTS
110 \inverteer dot
120 :LL10LDA@2;STA#5E;JMP(#3FE);]N.;P.$6
130 CLEARC;?S=A.R.%10+C+1;PLOT 13,0,(?S-1)
140 FORI=1TO B;DOS?I=S?(I-1)+R.%2;U.S?I<20
150 PLOT 13,I,(S?I-1);N.;REM TEKEN GRONDLIJN
160 FORI=0TO H-1;L?I=A.R.%(B+1);Q?I=I+S?(L?I)
170 PLOT 13,(L?I),(Q?I);N.
180 FORI=0TO H-1;L?I=A.R.%(B+1);BEEP R.&15+5,3
190 IFS?I>=H G.130
200 Q?I=H;PLOT 13,(L?I),H;LI.LL0;N.;G.180
210
220 Jan Wijnen
```

```
100REM INT-KLOK
110 PASS0;GOS.a;PASS1;GOS.a
120 END
130
140a ASM-BEGIN
150 .CODE #3300
160 .TABLE #7000,#8000
170 .LIST .OPTION #50
180
190:INTVEC=#204
200:LATCHLOW=#B806
210:LATCHHIGH=#B807
220:TIMERMODE=#B80B
230:INT'MASK=#B80E
240:TIMER1=#B804
250:INTTEL=#80
260:SECOND=#86
270:SECOND10=#85
280:MIN=#84
290:MIN10=#83
300:UREN=#82
310:UREN10=#81
320:CYFERS=#80
330:SCHERM=#8017
340
350:init
360 LDA@(int)%256 ;STA INTVEC
370 LDA@(int)/256 ;STA INTVEC+1
380 LDA@16 ;STA LATCHLOW
390 STA TIMER1
400 LDA@39 ;STA LATCHHIGH
410 STA TIMER1 + 1
420 LDA@#40 ;STA TIMERMODE
430 LDA@#C0 ;STA INT'MASK
440 CLI
450 RTS
460
470:int
480 LDA TIMER1
490 TXA ;PHA ;TYA ;PHA
500 DEC INTTEL ;BNE return
510 LDA@100 ;STA INTTEL
520 INC SECOND ;LDA SECOND
530 CMP@10 ;BMI display
540 LDX@0 ;STX SECOND
550 INC SECOND10 ;LDA SECOND10
560 CMP@6 ;BMI display
570 STX SECOND10
580 INC MIN;LDA MIN
590 CMP@10 ;BMI display
600 STX MIN
610 INC MIN10 ;LDA MIN10
620 CMP@6 ;BMI display
630 STX MIN10
640 INC UREN ;LDA UREN
650 CMP@4 ;BMI display
660 LDA UREN10 ;CMP@2 ;BNE dag
670 STX UREN ;STX UREN10;BEQ display
680:dag
690 LDA UREN ;CMP@10 ;BMI display
700 STX UREN ;INC UREN10
710:display
720 LDX@6 ;LDY@8
730:cyferlus
740 LDA CYFERS,X ;ORA@#30
750 CPX@4 ;BNE nietminuut
760 DEY
770:nietminuut
780 CPX@2 ;BNE nietuur
790 DEY
800:nietuur
810 STA SCHERM,Y
820 DEX ;DEY ;BNE cyferlus
830 LDA@ CH":"
840 STA SCHERM+3 ;STA SCHERM+6
850:return
860 PLA ;TAY ;PLA ;TAX
870 PLA
880 RTI
890
900 .SYMBOL
910 .END ASSEMBLY
920RETURN
```



EXEC.STAT

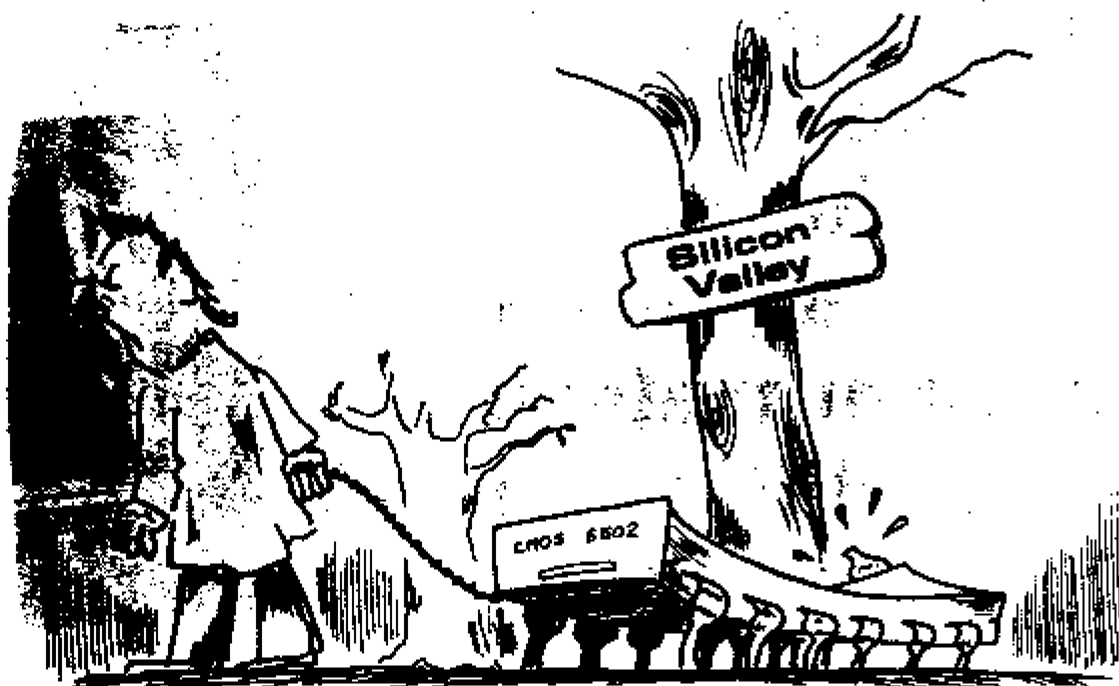
M.b.v. het volgende programma kunnen voortaan ook de niet-DOS-bezitters gebruik maken van het EXECute commando. Dit commando wordt op de bekende manier als uitbreiding aan P-CHARME geknoopt. EXEC zorgt ervoor dat de computer niet naar het toetsenbord kijkt als invoer medium, maar naar een file, die we ergens in het geheugen hebben gezet. In deze file kunnen dan bijv. allerlei commando's staan die anders alleen in direct mode worden geaccepteerd, of invoerwaarden waar een programma om vraagt bij uitvoering van INPUT-statements.
Dit commando EXEC heeft bovendien het voordeel dat de te "executeren" file gewoon mag worden gemaakt alsof het een BASIC-programma is, dus met regelnummers, iets dat met *EXEC uit DOS niet zonder meer mogelijk is. Als verdere verbetering is de mogelijkheid ingebouwd om de controle voor invoer tijdelijk terug te geven aan het toetsenbord. Wanneer deze invoer wordt afgesloten met RETURN wordt verder weer naar de file gekeken i.p.v. naar het toetsenbord. Dit wordt bereikt door een regel met @ in de te executeren file op te nemen.

Voorbeeld van zo'n file:

```
10 PLOETER
20 12.6
30 15
40 @
50 *LOAD"MEETWAARDEN"
60 PRINT $7$7$7
70 GRAFIEK
```

In regel 10 wordt PROGRAM PLOETER ge-RUN-d. Dit programma vraagt kennelijk 3 keer om invoer. De eerste keer wordt er 12.6 ingevoerd, de tweede keer 15. Voor de invoer van het derde getal of string wordt het commando teruggegeven aan het toetsenbord (regel 40). Vervolgens wordt een data-file "MEETWAARDEN" geladen, drie piepjes gegeven om te waarschuwen dat de recorder weer uit mag en in regel 70 wordt dan nog PROGRAM GRAFIEK ge-RUN-d. Misschien een onzinnig voorbeeld, maar hopelijk wordt de bedoeling duidelijk. Wanneer deze file bijv. op #9800 begint kan het executeren worden gestart met EXEC #98.

Hans Hegt



WIE ZEGT ER DAT DE ATOM NIET INTERNATIONAAL IS ???

```
10 PROGRAM EXEC.STAT
20
30 DIM LL9;F.I=0TO9;LLI=#FFFF;N.
40 Z=#9A;REM ZERO-PAGE ADRESSEN Z EN Z+1: WIJZER NAAR TEKST
50 H=#9FF0;REM ADRESSEN H EN H+1: OPSLAG OUDE RDCH-VECTOR
60 F=#9FF2;REM ADRES F:VLAG=0 FILE-,VLAG=#FF KEYBOARD-MODE
70 U=#9FF3;V=#9FF4; REM TIJDELIJKE OPSLAG X- EN Y-REGISTER
80 P.$21;P=A;GOS.a
90 P=A;GOS.a;P.$6
100 $T="EXEC";P.$T;T=T+LENT
110 ?T=LL0/256|#80;T?1=LL0&256;T?2=#80;T=T+2
120 A=P;T!1=A
130 END
140
150a[
160:LL0;JSR#C4E1;LDX#4;DEX;LDA#16,X;STX#4;STA Z+1;LDA@0;STA Z
170STA F;LDA#20A;STA H;LDA#20B;STA H+1
180LDA@(LL1&256);STA#20A;LDA@(LL1/256);STA#20B;JMP#C55B
190:LL1;PHP;STX U;STY V;LDA F;CMP@#FF;BEQLL4
200:LL2;LDX@0;LDA(Z,X);INC Z;BNELL3;INC Z+1
210:LL3;CMP@#0D;BEQLL5;CMP@#60;BNELL8
220LDA@#FF;STA F;JSR#FD1A
230:LL4;JSRLL9;CMP@#0D;BNELL8;LDY@0;STY F;JMP LL2
240:LL5;LDX@0;LDA(Z,X);CMP@#FF;BEQLL6;CLC;LDA@2;ADC Z;STA Z
250LDA@0;ADC Z+1;STA Z+1;JMPLL7
260:LL6;LDA H;STA#20A;LDA H+1;STA#20B
270:LL7;LDA@#0D
280:LL8;PLP;LDX U;LDY V;RTS
290:LL9;JMP(H)
300]
310 RETURN
```

```
  10 REM LAATSTE REDMIDDEL
  20 REM VOOR DE MDCR
  30
  40 DIM LL20
  50 V=#B800;REM VIA ADRES
  60 Q=#82;REM BUFFER ADRES
  70 M=#80;REM DRIE ADRESSEN ZERO PAGE
  80 P=#3C00;P.$21;GOS.a;P=#3C00;GOS.a;P.$6
  90 LINKLL0;REM INIT VIA
 100 REM SHIFT IN --> REWIND TAPE
 110 IF?#B001<255 V?1=8;DOU.V?1&64
 120 CLEAR4
 130 V?1=4;LINKLL4;V?1=0
 140 REM COPIEER BUFFER NAAR WERKELIJK ADRES
 150 P.$12;P." COPY #820A,#860A,#...."'
 160 END
 170a[
 180 :LL0 SEI;LDA@#3C;STA V+3;LDA@0;STA V+6
 190 LDA @Q;STA M+1;LDA@0;STA M
 200 RTS
 210\
 220\ WACHT OP PREAMBLE #AA
 230\
 240 :LL1 LDA@0;STA M+2
 250 :LL5 LDA@2
 260 :LL6BIT V+#D;BEQLL6
 270 LDA V+1;LSRA;ROL M+2;LDA M+2;CMP@#AA;BNE LL5
 280\
 290\ PREAMBLE GEVONDEN
 300\ LEES BLOK VAN TAPE
 310\
 320 :LL7 LDY@8
 330 :LL2 LDA@2
 340 :LL3 BIT V+#D;BNE LL8
 350\
 360\ BEINDIGING BIJ SHIFT IN
 370\
 380 BIT#B001;BMILL3;RTS
 390\
 400\ SCHUIF GELEZEN BIT IN BYTE
 410\
 420 :LL8 LDA V+1;LSRA;ROL M+2
 430 DEY;BNE LL2
 440\
 450\ ZET GELEZEN BYTE IN BUFFER
 460\
 470 LDA M+2;STA(M),Y
 480 INC M;BNE LL4;INC M+1
 490 :LL4 JMP LL7
 500];R.
```

DIT IS EEN PROGRAMMA DAT EEN BLOK VAN TAPE KAN LEZEN ALS HET MDCR PROGRAMMA NIET MEER IN DE COMPUTER BESCHIKBAAR IS.

GEBRUIK :
RUN HET PROGRAMMA
ALS SHIFT INGEDRUKT BLIJFT DAN WORDT DE TAPE EERST TERUGGESPOELT.
DAN WORDT DE DATA IN DE BUFFER GELADEN TOT WEER SHIFT INGEDRUKT WORDT.
HET BESTE KAN JE HET BLOK VOOR BLOK INLEZEN EN IN HET GEHEUGEN COPIEREN.

JAN WIJNEN
KEMPENSEBAAN 18
5613 JD EINDHOVEN
040 - 455852

```
 10 PROGRAM BINNENSTEBUITEN
 20 P.$21;F.I=1TO2
 30[.BA#3C00;.TA#9800
 40:START LDA@#00;STA#C0
 50       LDA@#80;STA#C1
 60:BEGIN LDA@#00;STA#C2
 70       LDA@#1F;STA#C3
 80:LOOP  LDY#C2
 90       LDA(#C0),Y
100       LDX@8
110:LOOP2 LSRA
120       ROL#C4
130       DEX
140       BNE LOOP2
150       LDY#C3
160       LDA(#C0),Y
170       LDX@8
180:LOOP3 LSRA
190       ROL#C5
200       DEX
210       BNE LOOP3
220       LDA#C4
230       STA(#C0),Y
240       LDY#C2
250       LDA#C5
260       STA(#C0),Y
270       INC#C2
280       DEC#C3
290       LDA#C2
300       CMP@#10
310       BNE LOOP
320       CLC
330       LDA#C0
340       ADC@#20
350       STA#C0
360       BCC JUMP1
370       INC#C1
380:JUMP1 LDA#C1
390       CMP@#98
400       BNE BEGIN
410       RTS
420]
430N.;P.$6;E.
```

Dit programma verwisselt in een CLEAR 4 scherm links en rechts met elkaar.
Het scherm wordt dus gespiegeld tov de lijn X=128.

Kees van Oss.

```
 10 PROGRAM ONDERSTEBOVEN
 20 P.$21;F.I=1TO2
 30[.BA#3C00;.TA#9800
 40:START LDA@#00;STA#C0
 50       LDA@#80;STA#C1
 60       LDA@#E0;STA#C2
 70       LDA@#97;STA#C3
 80:LOOP  LDY@0
 90:BEGIN LDA(#C2),Y
100       PHA
110       LDA(#C0),Y
120       STA(#C2),Y
130       PLA
140       STA(#C0),Y
150       INY
160       CPY@32
170       BNE BEGIN
180       CLC
190       LDA#C0
200       ADC@#20
210       STA#C0
220       BCC JUMP1
230       INC#C1
240:JUMP1 SEC@#20
250       LDA#C2
260       SBC@#20
270       STA#C2
280       BCS JUMP2
290       DEC#C3
300:JUMP2 LDA#C3
310       CMP@#8B
320       BNE LOOP
330       RTS
340]
350N.;P.$6;E.
```

Met dit programma kun je razend-snel een CLEAR 4 scherm ondersteboven zetten.
Het scherm wordt gespiegeld tov de lijn Y=96.

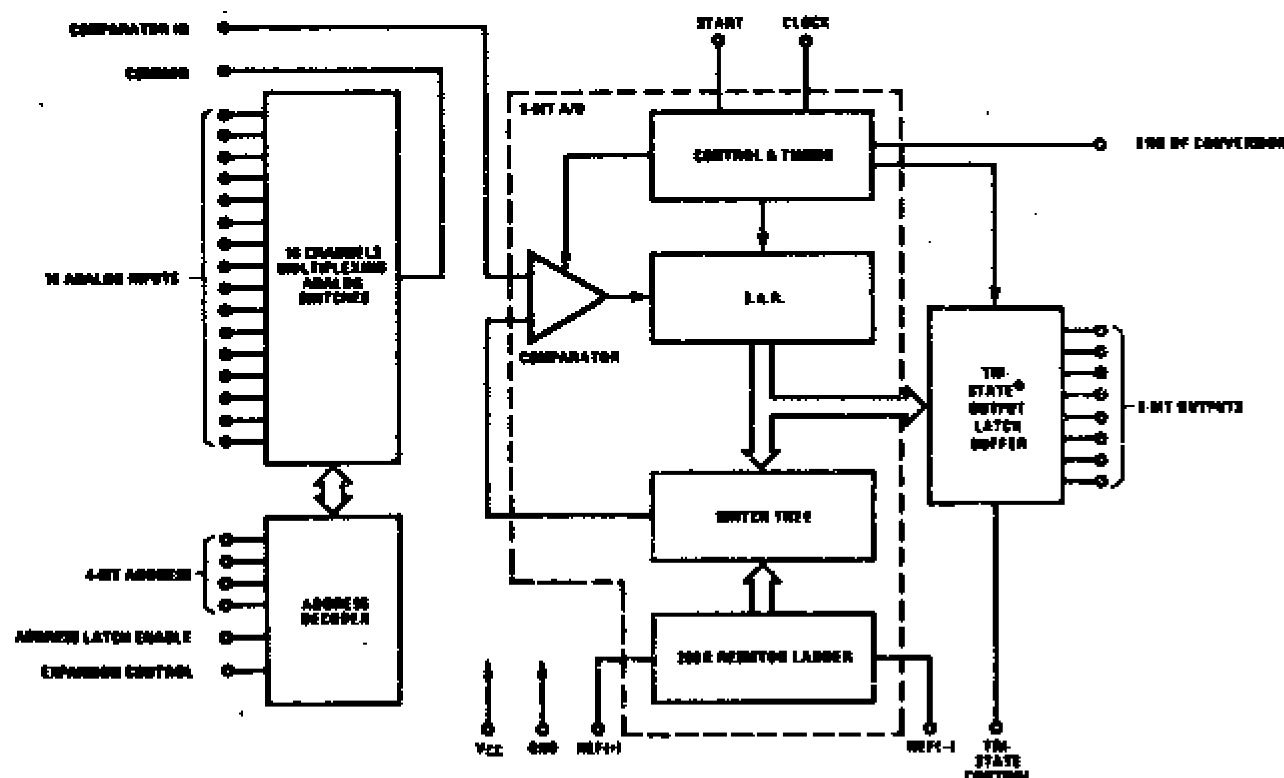
Kees van Oss.

EEN EENVOUDIGE ANALOOG DIGITAAL CONVERTER

In de krant van een paar weken terug las ik een stukje over een dumpzaak van electronica onderdelen. Meteen erop af natuurlijk. Na wat gesnuffel in oude gokkasten kwam ik een ADC0816 tegen. Hier volgt het relaas van mijn pogingen het ding aan de praat te krijgen.

De ADC0816 is een CMOS IC dat een 8 bits AD converter, een 16 kanaals analoge multiplexer en direct op een microprocessor aan te sluiten data- adres- en control lijnen bevat.

Block Diagram



ADC0816

Bij een klok frequentie van 640 kHz is de omzettingstijd 100 microseconden. De tijd bij 1 MHz is dan 100*640/1000=64 microseconden. Volgens de specificatie is de maximum klok frequentie 1200 kHz=1.2 MHz. Uitkijken dus met de 1.8 of 2 MHz versies van de Atom.
De AD omzetting gebeurt met behulp van een 256 elements weerstands netwerk. Het resultaat wordt in een latch met een tri-state uitgang geplaatst.

Timing Diagram

ADC0816

Dit is een mogelijke aansluiting aan de Atom bus:

NRDS

bv. #BC00

NWRS

1/2
74LS02

ϕ2

RDY

Typical Application

Dual-In-Line Package

TOP VIEW

Connection Diagram

Door het schrijf signaal wordt het adres voor de multiplexer van de adres bus ingeklokt, en de conversie gestart. Het EOC signaal gaat laag tot de conversie beindigt is. De EOC lijn kan aangesloten worden op een interrupt ingang, of op de RDY (ready) of SO (set overflow) ingang van de 6502. Een extra mogelijkheid wordt geboden door de EOC te verbinden met Start. Dan zal na de eerste startpuls de ADC continu actief blijven.

Een programma dat gebruikt maakt van de ADC op adres #BC00 en de EOC aangesloten op RDY ziet er zo uit:

```
10 B=#BC00
20 P.$12;IN."WELK KANAAL MOET GESCAND WORDEN "K
30 CLEAR4
40 FOR X=0 TO 255
50  B?K=0;REM START DE CONVERSIE VAN KANAAL K
60  Y=?B;REM LEES RESULTAAT
70  PLOT 13,X,Y
80 NEXT X
90 GOTO 20
```

Sluit maar eens de uitgang van een radio aan !

Gezien de mogelijkheden van dit IC hoop ik er nog veel plezier van te hebben. Daarom heb ik dit ook aan jullie doorgegeven.

Jan Wijnen
Brabant Oost

#BFFF	D	C (IC8 5)	B (IC8 2)	A (IC8 1)	Block	
0	0	0	0	0	0	RAM
1	0	0	0	1	1	RAM
2	0	0	1	0	2	EPROM 1
3	0	0	1	1	3	EPROM 1
4	0	1	0	0	4	EPROM 2
5	0	1	0	1	5	EPROM 2
6	0	1	1	0	6	EPROM 2
7	0	1	1	1	7	EPROM 2

Tabel 1

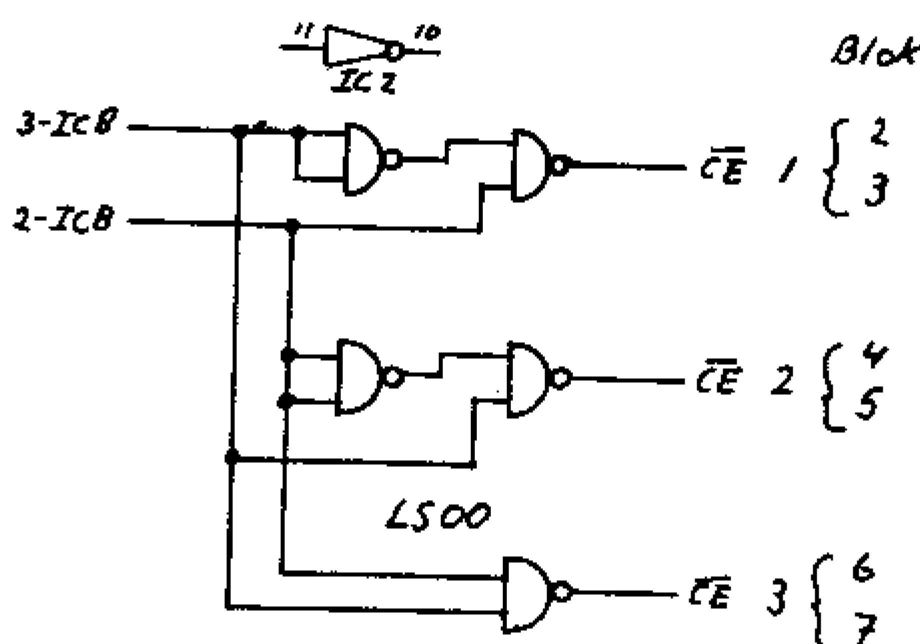


Fig 3

Eproms 2764 en/of 27128 op de schakelkaart.

Nu de 8 en 16k eproms zo goedkoop zijn geworden is het interessant om eens te kijken of die ook op de schakelkaart gebruikt kunnen worden. En jawel dat kan. In tabel 1 is de waarde van #BFFF decimaal en binair weergegeven. Tevens het bijbehorende blok. De binaire uitcodering van de CE is met de rechthoeken aangegeven. Uitgangen 2 en 3 van IC8 zijn nodig voor CE van eprom 1 (2764) terwijl uitgang 1 aan A12 schakelt tussen blok 2 en 3. Uitgang 3 is nodig voor CE van eprom 2 (27128) terwijl uitgangen 1 en 2 aan resp. A12 en A13 het schakelen tussen de blokken 4,5,6 en 7 verzorgen.
Fig.1 geeft de aansluiting van de 2764 met poorten 1 en 2 (LS00) voor de uitcodering van CE aan pen 20. De eprom wordt in een tussenvoet in een van de voeten van de schakelkaart geprikt. Dan zijn al 23 aansluitingen gemaakt, uitgaande van het gegeven dat de voeten op de kaart zijn omgebouwd voor het type 2732. De pennen 1,26,27 en 28 van de tussenvoet worden aan elkaar gesoldeerd en liggen via pen 26(24) aan de +5. Pen 2 komt aan pen 1-IC8.
Fig.2 geeft de aansluiting van de 27128 met poort 3 (LS00) voor CE aan pen 20. Pennen 1,27 en 28 worden aan +5 gelegd. Pen 2 aan 1-IC8 en pen 26 (uitgebogen) aan 2-IC8. De LS00 kan gemonteerd worden op IC8.
Er zijn nog twee mogelijkheden n.l. twee 2732's en een 27128 eprom. De 4k eproms komen dan normaal op de blokken 2 en 3, er hoeft dan niets gewijzigd te worden, de 27128 komt dan op de blokken 4,5,6 en 7 volgens fig.2. Voor poort 3 kan dan de nog vrije inverter (LS32) van IC2 gebruikt worden. Pen 11 aan 3-IC8 en pen 10 aan pen 20 van 27128.
De andere mogelijkheid is drie 2764's op de blokken 2 en 3, 4 en 5, 6 en 7. Zie fig.3. Er is dan ook maar een LS00 nodig als de vrije inverter (LS32) van IC2 gebruikt wordt. De eproms kunnen in elke willekeurige voet geprikt worden mits de bijbehorende CE lijn losgemaakt wordt van IC8 (pen buiten de voet buigen).

Sjaak Geene.

P.S. Het bovenstaande geldt voor de oude schakelkaart. Voor de nieuwe moet het ook wel mogelijk zijn.

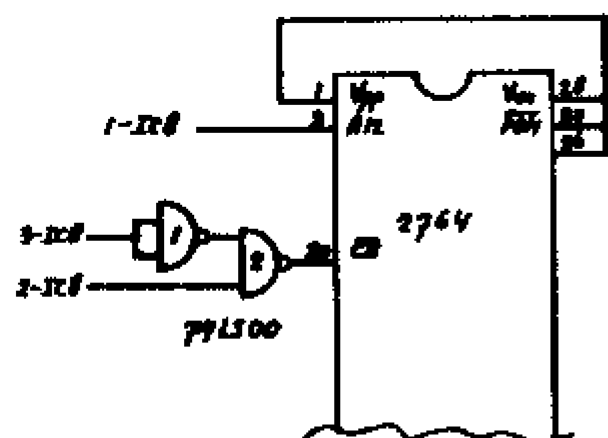


Fig 1

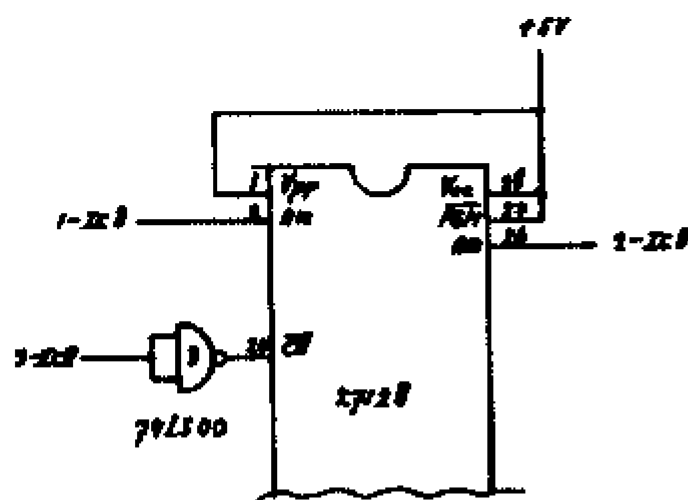


Fig 2